P.1877/80

8 1990

informatyka

Struktury systemów baz danych
Optyczna pamięć dyskowa
Wzmacnianie obrazów telewizyjnych

Cena 5800 zł

informatyka

Miesięcznik
ISSN 0542-9951. INDEX 36124

nr 8 1990
Sierpień
Rok wydania XXV

W numerze

	Strona
Fizyczne struktury systemów baz danych (1) – *Miłosz Muszyński*	1
Przemysłowe systemy czasu rzeczywistego (1). Rola i funkcje oprogramowania systemowego – *Mariusz Postół*	7
Optyczna pamięć dyskowa – *Jan Ryżko, Zdzisław Wrzeszcz*	10
Odwzorowania w relacyjnej bazie danych obiektów o nieokreślonych a priori typach – *Witold Rekuć*	15
System autorski JU-LEK – *Witold R. Rudolf*	18
Wzmacnianie obrazów telewizyjnych metodami komputerowego przetwarzania obrazów – *Nguyen Kim Sach*	22
System baz danych CX-DMOS (2) – *Lubomir Jurczak, Wojciech Warski*	24
Ze świata	28
Komputery do obliczeń naukowych Technologie układów scalonych w latach dziewięćdziesiątych	
Nowe książki	31
zBITki	
ON Wirus GOTO Gdziekolwiek	III okł.

W najbliższych numerach:

- Mieczysław Kłopotek opisuje Medytor – opracowany w Instytucie Podstaw Informatyki Polskiej Akademii Nauk – mówiący edytor języka polskiego.
- Piotr Żychowski przedstawia poszukiwania oraz wyniki stosowania algorytmu realizującego doprowadzenie słów tekstu języka polskiego do ustalonej formy gramatycznej.
- Krzysztof Zieliński rozważa możliwości rozproszenia przetwarzania zadań graficznych oraz przedstawia ich wykorzystanie w systemach graficznych budowanych przy użyciu transputerów.
- Zdzisław Wrzeszcz charakteryzuje język Lisp oraz naświetla genezę i podaje chronologię prac nad sztuczną inteligencją.

KOLEGIUM REDAKCYJNE:

mgr Jarosław DEMINET,
dr inż. Wacław ISZKOWSKI,
mgr Teresa JABŁOŃSKA,
(sekretarz redakcji)
Władysław KLEPACZ
(redaktor naczelny)
dr inż. Wojciech MOKRZYCKI,
mgr inż. Jan RYŻKO,
mgr Hanna WŁODARSKA.

PRZEWODNICZĄCY
RADY PROGRAMOWEJ:

Prof. dr hab.
Juliusz Lech KULIKOWSKI

WYDAWCA: Wydawnictwo Czasopism i Książek Technicznych SIGMA NOT
Spółka z o.o.
ul. Biała 4,
00-950 WARSZAWA
skrytka pocztowa 1004

Redakcja: 01-552 Warszawa
Pl. Inwalidów 10, p. 128, 136
tel. 39-14-34

Materiałów nie zamówionych redakcja nie zwraca

DRUK:
PRASOWE ZAKŁADY GRAFICZNE
RSW „PRASA-KSIĄŻKA-RUCH"
ul. Dworcowa 13,
85-950 BYDGOSZCZ
zam. 1821/90
Obj. 4,0 ark. druk. Nakład 5250 egz.

Cena egzemplarza 5800 zł

W sprawach ogłoszeń prosimy zwracać się bezpośrednio do Redakcji

WARUNKI PRENUMERATY

Prenumeratorzy zbiorowi – jednostki gospodarki uspołecznionej, instytucje i organizacje społeczne zamawiają prenumeratę dokonując wpłat wyłącznie na blankiecie „wpłata-zamówienie" (jest to „polecenie przelewu" rozszerzone dla potrzeb Wydawnictwa o część dotyczącą zamówienia). Blankiety te będą dostarczane dotychczasowym prenumeratorom przez Zakład Kolportażu. Nowi prenumeratorzy otrzymają je po zgłoszeniu zapotrzebowania (pisemne lub telefoniczne) w Zakładzie Kolportażu, w Radach Wojewódzkich NOT bądź w Redakcjach czasopism.

Prenumeratorzy indywidualni – osoby fizyczne zamawiają prenumeratę dokonując wpłaty w UPT lub NBP na blankiecie NBP. Na odwrocie wszystkich odcinków blankietu należy wpisać tytuł czasopisma, okres prenumeraty, liczbę zamawianych egzemplarzy oraz wartość wpłaty. Wpłacać należy na konto: **Państwowy Bank Kredytowy III/O Warszawa nr 370015-7490-139-11.**

Prenumerata ulgowa – przysługuje wyłącznie osobom fizycznym – członkom SNT, studentom i uczniom szkół zawodowych. Warunkiem prenumeraty ulgowej jest poświadczenie blankietu wpłaty (przed jej dokonaniem) na wszystkich odcinkach pieczęcią Koła SNT, wyższej uczelni lub szkoły. Sposób zamawiania prenumeraty ulgowej jest taki sam jak prenumeraty indywidualnej. W prenumeracie ulgowej można zamówić tylko po jednym egzemplarzu każdego czasopisma.

Uwaga! Miesięcznik „Aura" może być zamawiany w prenumeracie ulgowej również przez uczniów szkół ogólnokształcących.

Prenumerata ze zleceniem wysyłki za granicę – zamawia się tak, jak prenumeratę indywidualną. Dodatkowo należy podać na blankiecie wpłaty nazwisko i dokładny adres odbiorcy.

Cena prenumeraty ze zleceniem wysyłki za granicę jest dwukrotnie wyższa.

Wpłaty na prenumeratę przyjmowane są w terminach:
do 10 listopada na każdy kwartał, I i II półrocze oraz cały rok następny,
do 28 lutego na II, III i IV kwartał oraz II półrocze,
do 31 maja na III i IV kwartał oraz II półrocze,
do 31 sierpnia na IV kwartał.

Zmiany w prenumeracie można zgłaszać pisemnie tylko w wyżej wymienionych terminach.

Informacji o prenumeracie udziela Zakład Kolportażu Wydawnictwa NOT SIGMA (ul. Bartycka 20, 00-716 Warszawa), skr. poczt. 1004, 00-950 Warszawa, tel. 40-00-21 wew. 248, 249, 293, 297, 299 lub 40-30-86 i 40-35-89.

Egzemplarze archiwalne czasopism można nabyć za gotówkę w Klubie Prasy Technicznej, Warszawa ul. Mazowiecka 12 (tel. 26-80-16), lub zamówić pisemnie. Zamówienia na egzemplarze archiwalne czasopism przyjmuje: Zakład Kolportażu, Dział Handlowy, 00-950 Warszawa, skr. poczt. 1004 (tel. 40-37-31), na rachunek dla instytucji lub za zaliczeniem pocztowym dla osób fizycznych.

Cena egzemplarza: normalna 5800 zł, ulgowa 1160 zł

Uwaga! Podane ceny mają charakter wstępny i mogą ulec zmianie, w związku z powyższym Wydawnictwo zastrzega sobie prawo żądania dopłat do już opłaconej prenumeraty.

P.18771/90
Biblioteka Główna Politechniki Poznańskiej

MIŁOSZ MUSZYŃSKI
Instytut Informatyki, Automatyki i Robotyki
Politechnika Poznańska

Fizyczne struktury systemów baz danych (1)

Bazy danych są nie tylko podstawą tradycyjnych systemów wyszukiwania i przechowywania informacji, ale także stanowią integralną część systemów ekspertowych, komputerowego wspomagania projektowania, robotyki, wizji komputerowej itd. Implementacja systemu bazy danych pociąga za sobą konieczność wyboru fizycznych struktur danych. Dla przeciętnego użytkownika bazy fizyczne struktury danych są zazwyczaj niewidoczne, posługuje się on bowiem opisem danych na poziomie logicznym. Fizyczne struktury danych mają jednak zasadniczy wpływ na najważniejszy czynnik determinujący przydatność użytkową systemu baz danych, czyli na jego efektywność.

Współczesne, komercyjne systemy baz danych umożliwiają zazwyczaj pewną ograniczoną ingerencję w fizyczną postać przechowywanych danych [3]. Ingerencja taka może być najczęściej dokonana jedynie przez administratorów bazy danych (ang. *database administrator*, DBA), którzy powinni mieć odpowiednie rozeznanie w dziedzinie fizycznych struktur danych. Rozeznanie takie jest także niezbędne przy dokonywaniu porównań różnych systemów baz danych, ich oceny i wyboru.

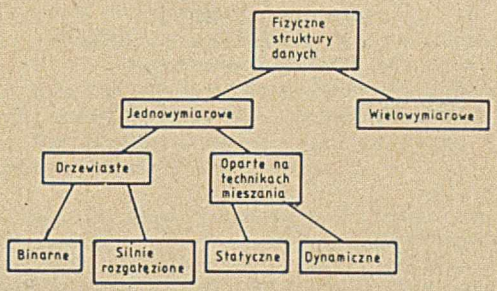
Niniejszy artykuł jest próbą wprowadzenia w dziedzinę fizycznych struktur systemów baz danych. Dokonana w nim klasyfikacja powinna ułatwić zorientowanie się w gąszczu różnych określeń i skrótów stosowanych przez producentów oprogramowania. Niektóre z omówionych struktur są zwiastunem najnowszych kierunków rozwoju obrazowych oraz obiektowych baz danych.

W systemach baz danych przyjmuje się najczęściej, że dane są reprezentowane w postaci zbiorów rekordów. Każdy rekord składa się z określonej liczby pól. Rekord reprezentuje zazwyczaj pewną umowną jednostkę informacji przechowywanej w bazie danych, natomiast pola tego rekordu reprezentują atrybuty tej jednostki (atrybutem pracownika jest nazwisko, imię, wiek, płaca, adres itd.). Najczęstszą operacją wykonywaną na danych jest wyszukiwanie rekordów na podstawie wartości niektórych lub wszystkich atrybutów, zwanych **argumentami wyszukiwania**. Atrybut rekordu, który zastosowany jako argument wyszukiwania pozwala go jednoznacznie zidentyfikować, jest nazywany **kluczem**. Rekordy mogą mieć wiele kluczy.

Podstawową funkcją struktur danych jest umożliwienie efektywnego dostępu do danych, czyli wykonywania trzech podstawowych operacji: wyszukiwania, wstawiania i usuwania rekordu. Najważniejszą z tych operacji jest wyszukiwanie rekordu, będące integralną częścią wszystkich pozostałych operacji dostępu. Od efektywnych struktur danych wymaga się jednak, aby umożliwiały nie tylko efektywne wyszukiwanie rekordów, lecz także ich wstawianie i usuwanie.

Wyszukiwanie rekordu jest również określane jako realizacja zapytań do bazy danych. Wyróżnia się dwa podstawowe rodzaje zapytań: zapytania punktowe i zapytania z zakresem. Zapytania punktowe dotyczą wyszukania jednego, ściśle określonego rekordu. Realizacja zapytań z zakresem wiąże się natomiast z wyszukaniem grupy rekordów spełniających określone w zapytaniu warunki. Struktura danych powinna umożliwiać efektywną realizację obu rodzajów zapytań.

Tradycyjne fizyczne struktury danych umożliwiają efektywny dostęp do danych wówczas, gdy argumentem poszukiwań jest jeden wybrany atrybut, najczęściej będący kluczem. Takie struktury są nazywane strukturami jednowymiarowymi. Struktury wielowymiarowe natomiast umożliwiają efektywny dostęp do danych według wielu atrybutów. Podział struktur danych na jedno- i wielowymiarowe stanowi podstawę przyjętej klasyfikacji struktur danych, przedstawionej na rys. 1. W niniejszej pierwszej części artykułu omówiono jednowymiarowe drzewiaste struktury danych. Część druga będzie poświęcona strukturom opartym na technikach mieszania, a trzecia – ostatnia – strukturom wielowymiarowym.

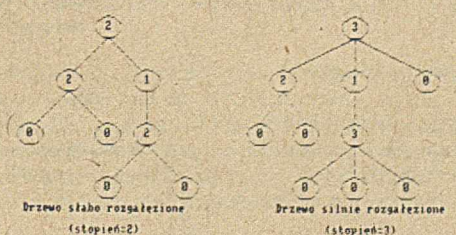


Rys. 1. Klasyfikacja fizycznych struktur baz danych

Spośród jednowymiarowych struktur danych największe znaczenie mają drzewa oraz struktury oparte na technikach mieszania (ang. *bashing*). Reprezentują one dwa odmienne sposoby podejścia do problemu przechowywania danych w pamięci komputera. W strukturze drzewiastej dane są umieszczone w sposób ściśle uporządkowany. W każdej chwili konkretny element danych może znajdować się tylko w jednym, dokładnie określonym miejscu struktury. W technikach mieszania zakłada się natomiast wiele możliwości umieszczenia konkretnego elementu danych, a odpowiednie procedury wyszukiwania uwzględniają wiele wariantów lokalizacji danego elementu.

Drzewo jest to skończony i niepusty zbiór, którego elementy są nazywane węzłami i spełniają dwa warunki [9]:
1) istnieje dokładnie jeden wyróżniony węzeł, zwany korzeniem drzewa,
2) pozostałe węzły znajdują się w m rozłącznych podzbiorach ($m >= 0$), z których każdy jest drzewem.
Powyższa definicja jest rekurencyjna, co odzwierciedla rekurencyjny charakter drzew i algorytmów dostępu do danych zawartych w strukturach drzewiastych.

Stopień węzła jest to liczba poddrzew tego węzła, a stopień drzewa – maksymalny stopień wszystkich węzłów tego drzewa. Drzewa o stopniu większym niż 2 nazywa się drzewami silnie rozgałęzionymi, a drzewa o stopniu równym dwa – drzewami słabo rozgałęzionymi lub binarnymi. Pojęcie stopnia drzewa i podział drzew zilustrowano na rys. 2.



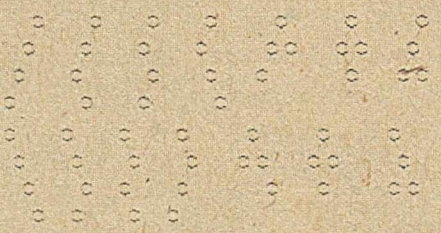
Rys. 2. Ilustracja pojęcia stopnia drzewa

Drzewa silnie rozgałęzione są najczęściej stosowane jako struktury danych przechowywanych w pamięci zewnętrznej (na przykład dyskowej); drzewa binarne i ich odmiany są natomiast stosowane w pamięci operacyjnej. Wiąże się to z odmiennymi kryteriami oceny struktur i algorytmów dostępu stosowanymi w pamięci operacyjnej i w pamięci zewnętrznej.

DRZEWA BINARNE

Drzewo binarne jest to skończony zbiór węzłów, który albo jest zbiorem pustym, albo zawiera korzeń oraz dwa drzewa binarne nazywane lewym i prawym poddrzewem tego korzenia.
Drzewo binarne nie jest równoznaczne z drzewem o stopniu równym 2, ponieważ:

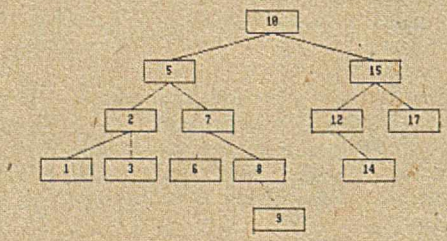
- drzewo binarne może być drzewem pustym,
- w drzewie binarnym rozróżnia się lewe i prawe poddrzewo, a więc drzewo złożone z korzenia i z lewego poddrzewa nie jest równoznaczne z drzewem złożonym z korzenia i prawego poddrzewa. Różnice te nie są jednak w praktyce istotne i dlatego przyjmuje się równoznaczność między drzewami o stopniu równym 2 i drzewami binarnymi. Na rysunku 3 pokazano 14 możliwych różnych drzew binarnych o czterech węzłach, wśród których znajdują się tylko cztery różne drzewa o stopniu równym 2.



Rys. 3. Drzewa binarne o czterech węzłach

W drzewie binarnym każdemu węzłowi przyporządkowuje się jeden rekord danych. Zazwyczaj w węzłach drzewa binarnego umieszcza się klucze identyfikujące rekordy oraz wskaźnik do rekordów. Innym rozwiązaniem jest przechowywanie w węzłach drzewa całych rekordów. Dla uproszczenia można założyć, że w węzłach drzewa binarnego znajdują się tylko klucze rekordów, abstrahując od szczegółów realizacji połączenia kluczy z rekordami.

Drzewo binarne przystosowane do wykonywania na nim operacji wyszukiwania, wstawiania i usuwania rekordów nazywa się drzewem poszukiwań binarnych. Musi ono spełniać następujące warunki [9]:
1) w każdym węźle znajduje się jedna wartość kluczy,
2) dla każdego węzła tego drzewa wszystkie klucze w lewym poddrzewie są mniejsze, a w prawym większe niż wartość klucza znajdującego się w tym węźle.
Rozmieszczenie wartości kluczy w drzewie poszukiwań binarnych, zgodne z drugim punktem jego definicji, gwarantuje możliwość zastosowania algorytmu poszukiwania równoważnego powszechnie stosowanej metodzie połowienia. Przykładowe drzewo poszukiwań binarnych przedstawiono na rys. 4.

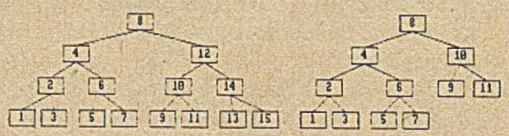


Rys. 4. Drzewo poszukiwań binarnych

Wspomniałem powyżej, że drzewa binarne są strukturą najczęściej stosowaną do danych w pamięci operacyjnej. W takiej sytuacji najważniejszym kryterium mającym wpływ na efektywność wykonywanych operacji jest liczba porównań potrzebnych do wyszukania konkretnego rekordu.

Wysokością drzewa nazywa się maksymalna liczba węzłów, które trzeba odwiedzić idąc od korzenia drzewa do któregoś z jego węzłów końcowych (liści). Wyszukiwanie w drzewie poszukiwań binarnych zawsze rozpoczyna się od korzenia. Jeżeli wartość argumentu wyszukiwania jest równa wartości klucza znajdującego się w korzeniu, to całe wyszukiwanie wymaga wykonania tylko jednej operacji porównania. W przeciwnym razie algorytm wyszukiwania jest wywoływany rekurencyjnie dla lewego lub prawego poddrzewa, w zależności od wyniku porównania. Algorytm wyszukiwania w drzewie poszukiwań binarnych można zapisać następująco:

1) jeżeli korzeń jest pusty, to zakończ wyszukiwanie i wyprowadź informację o nieznalezieniu klucza równego argumentowi wyszukiwania;
2) jeżeli korzeń nie jest pusty, to porównaj wartość klucza w korzeniu z argumentem wyszukiwania;
2. 1) jeżeli (argument < klucz), to wywołaj rekurencyjnie algorytm dla lewego poddrzewa;
2. 2) jeżeli (argument > klucz), to wywołaj rekurencyjnie algorytm dla prawego poddrzewa;
2. 3) jeżeli (argument = klucz), to zakończ wyszukiwanie i wyprowadź informację o znalezieniu klucza równego argumentowi wyszukiwania.



Rys. 5. W pełni zrównoważone drzewa poszukiwań binarnych

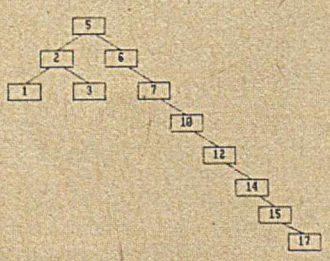
Wysokość drzewa binarnego determinuje liczbę porównań potrzebnych do odnalezienia konkretnego rekordu. Przy danej liczbie rekordów minimalna wysokość drzewa binarnego implikuje maksymalną efektywność wykonywanych operacji. Drzewo binarne o najmniejszej wysokości przy zadanej liczbie węzłów (rekordów) nazywa się drzewem binarnym w pełni zrównoważonym (rys. 5). Formalnie drzewo binarne nazywa się drzewem w pełni zrównoważonym, jeśli wszystkie jego liście znajdują się na co najwyżej dwóch sąsiednich poziomach [4]. W pełni zrównoważone drzewa binarne umożliwiają dostęp do danych w czasie $0(\log_2 n)$. Drzewa binarne w pełni zrównoważone stanowią specyficzny przypadek drzew binarnych. Dysponując zbiorem n rekordów, których klucze są uporządkowane monotonicznie, można skonstruować drzewo poszukiwań binarnych na wiele sposobów. Rekurencyjny wzór na liczbę różnych drzew poszukiwań binarnych o n węzłach jest następujący [8, 9]:

$$b_n = \sum_{k=0}^{n-1} b_k * b_{n-k-1}$$

gdzie b_i oznacza liczbę różnych drzew binarnych o i węzłach. Powyższy wzór można interpretować jako stwierdzenie faktu, że drzewo binarne o n węzłach składa się z korzenia, lewego poddrzewa o k ($k < n$) węzłach oraz prawego poddrzewa o $n-k-1$ węzłach. W wyniku rozwiązania powyższego równania metodą funkcji tworzącej, wzór na liczbę różnych drzew binarnych o n węzłach przyjmuje postać:

$$b_n = \frac{1}{n+1} \cdot \binom{2n}{n} = \frac{4^n}{\Pi^{1/2} \cdot n^{3/2}}$$

Zgodnie z powyższym wzorem istnieje 14 różnych drzew binarnych o czterech węzłach (rys. 3). Liczba różnych drzew binarnych rośnie bardzo szybko ze wzrostem liczby węzłów n. Już dla $n = 15$ istnieje 9 694 845 różnych drzew binarnych, z których zaledwie jedno jest w pełni zrównoważone, wiele jest natomiast drzew zwyrodniałych, tj. takich, w których czas poszukiwania jest rzędu $0(n)$ zamiast $0(\log_2 n)$ (rys. 6).



Rys. 6. Zwyrodniałe drzewo poszukiwań binarnych

Mając dany statyczny zbiór n rekordów zawsze można zbudować w pełni zrównoważone drzewo binarne zapewniające najefektywniejszy

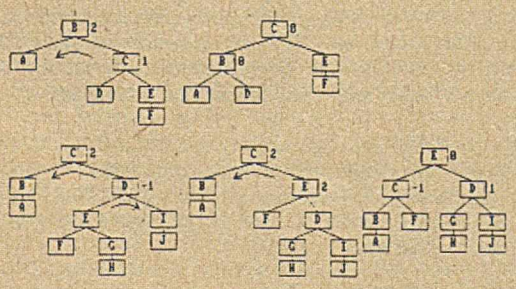
dostęp do danych znajdujących się w tym zbiorze. W bazach danych najczęściej zbiory są dynamiczne, w których często wykonuje się operacje wstawiania i usuwania rekordów, co powoduje utratę pełnego zrównoważenia drzewa binarnego. Próbowano stworzyć efektywne algorytmy zapobiegające wyrodnieniu drzew bez konieczności zmiany struktury całego drzewa. Algorytmów takich nie znaleziono. W praktyce okazało się, że jedynym sposobem utrzymania drzewa w pełni zrównoważonego jest okresowa pełna reorganizacja drzewa, równoznaczna pod względem kosztów z budową nowego drzewa poszukiwań binarnych.

Ze względu na istotne zalety drzew binarnych próbowano pogodzić się z niemożnością utrzymania pełnego zrównoważenia tych drzew. Obliczono, że przy równym prawdopodobieństwie wszystkich permutacji kolejności operacji wstawiania poszczególnych rekordów do drzewa binarnego jego wysokość będzie średnio tylko 1. 386 raza większa niż wysokość odpowiadającego mu drzewa w pełni zrównoważonego [2, 9]. W praktyce jednak założenie o równym prawdopodobieństwie wszystkich permutacji nie jest spełnione. Bardzo często zdarzają się monotoniczne ciągi rekordów, których wstawianie powoduje lawinowe zwiększanie się wysokości drzewa i w konsekwencji jego wyrodnienie.

Początkowo próbowano rozwiązać problem drzew zwyrodniałych za pomocą specjalnych procedur okresowo sprawdzających wysokość drzewa i dokonujących reorganizacji w przypadku przekroczenia przez drzewo pewnej określonej wysokości. Doprowadziło to z czasem do powstania koncepcji drzewa słabo lub nie w pełni zrównoważonego. Drzewo takie, nazywane drzewem zrównoważonym, jest wynikiem kompromisu między bardzo kosztownym w utrzymaniu drzewem w pełni zrównoważonym a niedopuszczalnym – ze względu na czasy dostępu – drzewem zwyrodniałym. Drzewo zrównoważone przyjęto nazywać AVL-drzewem. Nazwa ta pochodzi od nazwisk twórców tego drzewa (Adelson-Welskij i Landis), a jego definicja jest następująca: *drzewem AVL nazywa się takie drzewo, w którym wysokość lewego poddrzewa każdego węzła różni się od wysokości prawego poddrzewa o nie więcej niż 1* [9].

Drzewa binarne w pełni zrównoważone są zatem AVL-drzewami o minimalnej wysokości, a AVL-drzewami o maksymalnej wysokości są tak zwane drzewa Fibonacciego [4].

Podstawową zaletą AVL-drzew jest istnienie efektywnych algorytmów umożliwiających ich utrzymanie podczas wstawiania i usuwania. Z każdym węzłem AVL-drzewa jest związany wskaźnik zrównoważenia węzła, będący różnicą wysokości jego lewego i prawego poddrzewa. W drzewie mającym własność AVL wskaźnik zrównoważenia wszystkich węzłów wynosi 1, 0 lub −1. Po wstawieniu lub usunięciu rekordu sprawdza się, czy wartość bezwzględna wskaźnika zrównoważenia któregoś z węzłów nie stała się większa niż 1, czyli czy własność AVL nie została naruszona. Jeżeli tak się stało, to dokonując jednego lub dwóch tak zwanych obrotów przywraca się stan, w którym jest zachowana własność AVL. Wszystkie możliwe przypadki konieczności modyfikacji AVL-drzewa można sprowadzić do dwóch przypadków pokazanych na rys. 7. Konieczność dokonania jednego obrotu zachodzi wówczas, gdy wskaźnik zrównoważenia sąsiednich dwóch węzłów, z których jeden ma wartość bezwzględną wskaźnika większą niż 1, a drugi jest jego bezpośrednim następnikiem od strony wstawianego węzła, mają ten sam znak. Konieczność dokonania podwójnego obrotu zachodzi wówczas, gdy węzły te mają różne znaki.



Rys. 7. Przywrócenie zrównoważenia AVL-drzewa za pomocą pojedynczego (u góry) i podwójnego (u dołu) obrotu

Doświadczalnie stwierdzono, że średnio w 2/3 wstawień nie występuje potrzeba dokonywania obrotów, w 1/6 występuje potrzeba dokonania pojedynczego obrotu i w 1/6 obrotu podwójnego [9]. Zarówno wstawienie, jak i usunięcie rekordu z AVL-drzewa ma złożoność $0(\log_2 n)$, zatem praktycznie, AVL-drzewa są najbardziej korzystnymi strukturami danych spośród drzew binarnych.

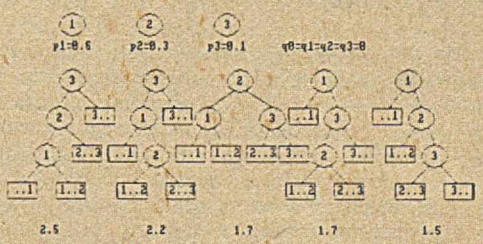
Zaletą drzew AVL jest bardzo efektywny dostęp do danych, nie wymagający żadnych obliczeń arytmetycznych, poszukiwań sekwencyjnych ani innych czasochłonnych operacji. Wadą są natomiast dość duże wymagania pamięciowe, na jeden klucz przypadają bowiem dwa wskaźniki. AVL-drzewo jest jedną z najczęściej stosowanych w pamięci operacyjnej struktur danych.

W powyższych rozważaniach zakładałem jednakowe prawdopodobieństwo wystąpienia każdej z możliwych wartości kluczy jako argumentu poszukiwania. W praktyce występują również takie sytuacje, w których można ustalić prawdopodobieństwo wystąpień wartości poszczególnych kluczy jako argumentów wyszukiwania. Można także ustalić prawdopodobieństwa wystąpień wartości kluczy z przedziałów odpowiadających tak zwanemu nieudanemu wyszukiwaniu, czyli wyszukiwaniu kluczy nie istniejących w drzewie poszukiwań. Przedziały odpowiadające wyszukiwaniu nieudanemu są reprezentowane zazwyczaj w postaci $n+1$ dodatkowych węzłów drzewa. Informacje takie mogą być wykorzystane w celu utworzenia optymalnego drzewa poszukiwań binarnych, w którym czas wyszukiwania danego klucza jest odwrotnie proporcjonalny do częstości wystąpień wartości tego klucza jako argumentu wyszukiwania.

Niech $p_1, p_2, \ldots, p_n$ oznaczają prawdopodobieństwa wystąpień kluczy $k_1, k_2, \ldots, k_n$ jako argumentów wyszukiwania, a $q_0, q_1 \ldots, q_n$ – prawdopodobieństwa wystąpień wartości kluczy z przedziałów odpowiadających dodatkowym $n+1$ węzłom $w_0, w_1, \ldots, w_n$ drzewa, reprezentującym poszukiwanie nieudane. Średni czas wyszukiwania w drzewie można wyrazić wzorem [1, 4, 8, 9]:

$$\sum_{i=1}^{n} p_i \cdot (1 + poziom(k_i)) + \sum_{i=0}^{n} q_i \cdot poziom(w_i)$$

gdzie *poziom(w)* oznacza długość drogi dzielącej węzeł w od korzenia drzewa. Powyższy wzór umożliwia obliczenie współczynników optymalności drzew binarnych o n węzłach i wybranie spośród nich drzewa optymalnego, czyli drzewa o najmniejszym średnim czasie wyszukiwania. W praktyce takie postępowanie nie jest stosowane ze względu na olbrzymią liczbę różnych drzew binarnych o i węzłach. Istnieją algorytmy dokładne umożliwiające konstruowanie optymalnych drzew binarnych w czasie $0(n^2)$. Ma ono oczywiście sens jedynie dla statycznych lub w przybliżeniu statycznych zbiorów danych.



Rys. 8. Dobór optymalnego drzewa poszukiwań binarnych przy znanym rozkładzie wartości argumentu wyszukiwania

Przykład zastosowania wzoru na średni czas wyszukiwania w drzewie binarnym przedstawiono na rys. 8. Znajdują się na nim wszystkie drzewa binarne o trzech węzłach wraz ze średnimi czasami wyszukiwania obliczonymi za pomocą powyższego wzoru dla danych $p_1, p_2, p_3, q_0, q_1, q_2, q_3$. Drzewem optymalnym jest to drzewo, dla którego średni czas wyszukiwania wynosi 1.5.

DRZEWA SILNIE ROZGAŁĘZIONE – ORGANIZACJA PLIKÓW DYSKOWYCH

Drzew silnie rozgałęzionych nie stosuje się dla danych w pamięci operacyjnej, mimo iż mają lepszy stopień wykorzystania pamięci, niż AVL-drzewa. Miarą wykorzystania pamięci może być stosunek liczby wskaźników do liczby kluczy w węźle drzewa, który dla drzew silnie rozgałęzionych wynosi $s/(s-1)$, a dla AVL-drzew – 2 (s oznacza stopień drzewa silnie rozgałęzionego). Ta niewątpliwa zaleta drzew silnie

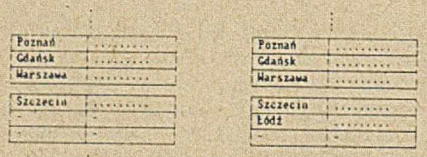
rozgałęzionych traci jednak na znaczeniu wobec mniejszej efektywności wyszukiwania oraz bardziej skomplikowanego wstawiania i usuwania. Efektywność wyszukiwania w drzewie silnie rozgałęzionym stopnia s jest rzędu $0(s \log_s n)$, co wynika z konieczności sekwencyjnego przeszukiwania węzłów drzewa.

Zalety drzew silnie rozgałęzionych ujawniają się dopiero w zastosowaniach dla danych w pamięci zewnętrznej, gdyż inne jest tu kryterium efektywności. Zakłada się, że operacja odczytu-zapisu jednego bloku pamięci zewnętrznej trwa wielokrotnie dłużej niż sekwencyjne przeszukiwanie tego bloku w pamięci operacyjnej. W związku z tym za kryterium efektywności przyjmuje się liczbę wykonanych operacji odczytu-zapisu bloków pamięci zewnętrznej. Biorąc to pod uwagę można stwierdzić, że drzewa silnie rozgałęzione są efektywniejsze niż drzewa binarne, mogą bowiem składać się z dużych węzłów, obejmujących całe bloki dyskowe, co pozwala minimalizować liczbę przesłań blokowych między pamięcią operacyjną a pamięcią zewnętrzną.

Większość znanych organizacji plików dyskowych opiera się na koncepcji drzew silnie rozgałęzionych. Dla pełnego omówienia zagadnienia przechowywania danych w pamięci zewnętrznej warto jednak wspomnieć o dwu organizacjach nie wykorzystujących struktur drzewiastych: organizacji stogowej i organizacji sekwencyjnej.

Organizacja stogowa

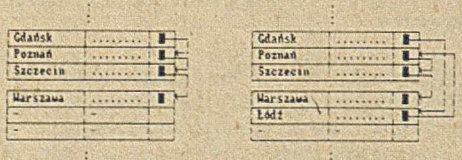
Organizacja stogowa (ang. heap) jest najprostszą ze znanych organizacji plików dyskowych [5, 10]. Rekordy są wstawiane do bloków pamięci dyskowej w sposób nieuporządkowany, a wyszukanie konkretnego rekordu wymaga dostępu średnio do połowy wszystkich bloków pliku, a w najgorszym przypadku – do wszystkich bloków pliku. Organizacja ta jest często oferowana jako opcja przez współczesne komercyjne systemy baz danych do przechowywania tymczasowych zbiorów danych, dla których nie opłaca się tworzenie struktur bardziej skomplikowanych [3]. Organizację stogową przykładowego pliku rekordów przed i po wstawieniu rekordu „Łódź" przedstawiono na rys. 9.



Rys. 9. Organizacja stogowa pliku dyskowego

Organizacja sekwencyjna

Organizacja sekwencyjna umożliwia efektywne przetwarzanie rekordów w porządku zgodnym z porządkiem klucza [5, 6]. W celu umożliwienia szybkiego wyszukiwania rekordów, każdemu rekordowi jest przyporządkowany wskaźnik do następnego rekordu. Istnienie wskaźników umożliwia wstawianie i usuwanie rekordów w pliku sekwencyjnym bez konieczności reorganizacji całego pliku. Najczęściej plik sekwencyjny inicjuje się w taki sposób, że fizyczne uporządkowanie rekordów odpowiada porządkowi logicznemu implikowanemu przez wartości wskaźników i jest zgodne z monotonicznym porządkiem wartości klucza. W wyniku kolejnych wstawień i usunięć rekordów fizyczne uporządkowanie przestaje odpowiadać porządkowi logicznemu, co powoduje zmniejszenie efektywności dostępu. Pliki sekwencyjne wymagają okresowej reorganizacji w celu utrzymania efektywności dostępu. Realizacja zapytań punktowych w pliku sekwencyjnym może wymagać dostępu średnio do więcej niż połowy wszystkich bloków pliku, jednak realizacja zapytań z zakresem jest o wiele efektywniejsza niż w plikach o organizacji stogowej. Organizację sekwencyjną pliku o zawartości zgodnej z plikiem z rys. 9 przedstawiono na rys. 10.



Rys. 10. Organizacja sekwencyjna pliku dyskowego

Obie organizacje należą do najmniej efektywnych organizacji plików dyskowych. Pozostałe organizacje stanowią połączenie koncepcji organizacji sekwencyjnej ze strukturą drzewa silnie rozgałęzionego, dzięki czemu ich efektywność jest wielokrotnie większa.

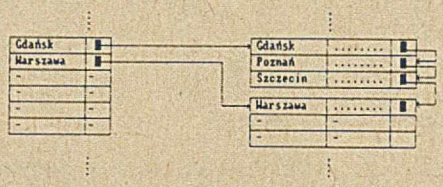
Organizacja indeksowo-sekwencyjna

Organizacja indeksowo-sekwencyjna stanowi połączenie organizacji sekwencyjnej z organizacją indeksową [5, 6, 10]. Pliki indeksowo-sekwencyjne umożliwiają zarówno sekwencyjne przetwarzanie całego pliku, jak i swobodny dostęp do wybranych rekordów pliku za pomocą indeksu. Indeksem nazywa się skojarzony z plikiem rekordów danych dodatkowy plik tak zwanych rekordów indeksowych, zawierających wartości kluczy rekordów pliku głównego oraz wskaźniki do rekordów identyfikowanych przez te klucze. Rozróżnia się dwa rodzaje indeksów:

- **gęsty**, w którym jeden rekord indeksowy przypada na jeden rekord danych,
- **rzadki**, w którym jeden rekord indeksowy przypada na grupę rekordów danych.

Indeks gęsty umożliwia efektywniejsze wyszukiwanie konkretnego rekordu, jednak wymaga znacznych zasobów pamięci. Indeks rzadki organizuje się najczęściej w taki sposób, aby jeden element indeksu rzadkiego przypadał na jeden blok rekordów danych. Wskazywane przez indeks rzadki bloki przesyła się do pamięci operacyjnej, gdzie dokonuje się ich przeszukiwania.

Jeśli indeks nie mieści się w jednym bloku pamięci dyskowej, to tworzy się indeks indeksu w taki sposób, aby indeks najwyższego poziomu mieścił się w jednym bloku pamięci dyskowej. Wielopoziomowa hierarchia indeksów tworzy zatem strukturę drzewa silnie rozgałęzionego, którego węzłami są bloki pamięci dyskowej. Stopień drzewa jest wyznaczony przez liczbę rekordów indeksowych przypadających na jeden blok dyskowy (rys. 11).



Rys. 11. Organizacja indeksowo-sekwencyjna pliku dyskowego

Plik indeksowo-sekwencyjny, podobnie jak plik sekwencyjny, charakteryzuje się zmniejszeniem efektywności dostępu w miarę wykonywanych wstawień i usunięć rekordów. W wyniku kolejnych wstawień i usunięć fizyczny porządek zarówno indeksu, jak i rekordów danych przestaje odpowiadać porządkowi logicznemu. Organizacja indeksowo-sekwencyjna z przyczyn historycznych jest często określana jako organizacja typu ISAM (ang. *indexed sequential access method*, indeksowo-sekwencyjna metoda dostępu stosowana w komputerze IBM 370). W chwili obecnej skrót ISAM używany jest w dwóch kontekstach: historycznym i współczesnym.

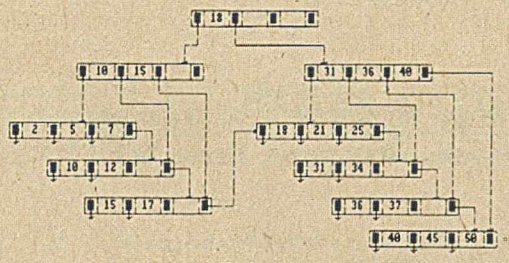
Historycznie ISAM oznacza metodę dostępu do plików sztywno związanych z określonym typem sprzętu [6]. W metodzie tej rekordy odpowiadają ścieżkom, bloki cylindrom, wyróżnione ścieżki zawierają indeks rekordów cylindra itd. Tak interpretowanej metodzie ISAM jest przeciwstawiana niezależna od sprzętu metoda wirtualnego dostępu, określana jako VSAM (ang. *virtual sequential access method*).

Współcześnie ISAM oznacza taką organizację pliku, w której struktura indeksu jest stała, co powoduje zmniejszenie efektywności dostępu w miarę kolejnych wstawień i usunięć rekordów pliku. Niektóre systemy baz danych oferują organizację typu ISAM jako opcję zalecaną dla danych o charakterze statycznym [3].

Organizacja typu B+ drzewo

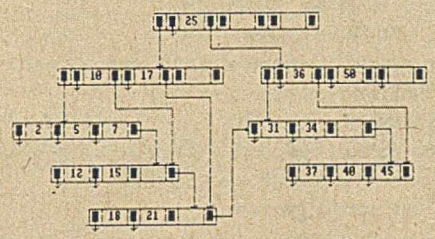
Najbardziej rozpowszechnioną organizacją pliku rekordów danych jest organizacja w postaci B+ drzewa (rys. 12). Plik w postaci B+ drzewa różni się od pliku ISAM sposobem organizacji indeksu. Do podstawowych cech indeksu typu B+ drzewo należą [5]:

- wysokość drzewa zmienia się dynamicznie, w zależności od liczby rekordów w pliku,
- węzły B+ drzewa są zawsze wypełnione minimum w połowie,
- wartości kluczy w węzłach nie będących liśćmi (węzłach pośrednich) oraz w liściach mogą się powtarzać,
- liście są połączone wskaźnikami tworząc strukturę sekwencyjną.



Rys. 12. Indeks o strukturze B+ drzewa

Charakterystyczną cechą B+ drzewa jest to, że do wykonywania operacji wstawiania i usuwania stosuje się specjalnie zmodyfikowane algorytmy, które mogą zmieniać wysokość B+ drzewa, a także mają własność utrzymywania odpowiedniego współczynnika wypełnienia węzłów drzewa. Przyjmuje się, że w węzłach B+ drzewa stopnia s może znajdować się od $\lceil s/2 \rceil$ do s wartości klucza. Jeśli w wyniku usunięcia liczba kluczy znajdujących się w węźle będzie wynosić mniej niż $\lceil s/2 \rceil$, to węzeł taki albo zostanie usunięty, a jego klucze przeniesione do innego węzła, albo część kluczy z innego węzła zostanie do niego przeniesiona w taki sposób, aby liczba kluczy we wszystkich węzłach była większa niż $\lceil s/2 \rceil$. Analogiczne działania towarzyszą operacji wstawiania. Węzły pośrednie B+ drzewa oraz liście mają identyczną postać. W każdym węźle znajduje się s wskaźników oraz $s-1$ wartości kluczy. Jedyną różnicą między węzłami pośrednimi a liśćmi jest to, że wskaźniki w węzłach pośrednich odnoszą się do innych węzłów B+ drzewa, a wskaźniki znajdujące się w liściach odnoszą się bezpośrednio do rekordów danych. Ponieważ jednak liczba potrzebnych wskaźników w liściach przekracza liczbę wartości kluczy o jeden, dodatkowy wskaźnik wykorzystuje się do połączenia liści B+ drzewa w strukturę sekwencyjną, co ułatwia realizację zapytań z zakresem.



Rys. 13. Indeks o strukturze B− drzewa

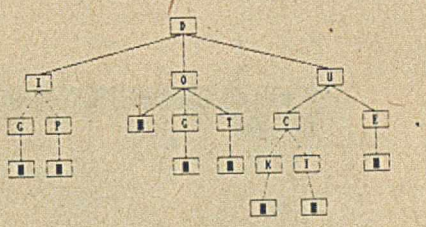
Algorytm realizujący zapytanie punktowe za pomocą indeksu w postaci B+ drzewa musi zawsze przebyć drogę o jednakowej długości od korzenia do któregoś z liści. B+ drzewo jest odmianą struktury o nazwie B− drzewo (rys. 13), która nie ma tej własności [4, 10]. W B drzewach nie występuje nadmiarowe przechowywanie tych samych wartości kluczy w węzłach pośrednich i liściach. Odnalezienie żądanego rekordu jest możliwe bezpośrednio po napotkaniu odpowiedniej wartości klucza w węźle pośrednim. Jest to jednak okupione takimi niedogodnościami, jak różne formaty węzłów pośrednich i liści oraz bardziej skomplikowane i przez to mniej efektywne algorytmy wstawiania i usuwania. Jedyną zaletą B− drzew w porównaniu z B+ drzewami jest mniejszy średni czas wyszukiwania ze względu na możliwość zakończenia wyszukiwania w węźle pośrednim. Zaleta ta traci jednak na znaczeniu dla dużych indeksów, w których dla ogromnej większości przypadków wyszukiwań jest konieczne przejście całej drogi od korzenia do liścia. Obecnie B drzewa rzadko się stosuje, natomiast pliki z indeksem w postaci B+ drzewa są oferowane przez niemal wszystkich producentów oprogramowania w dziedzinie baz danych [3].

Niektóre systemy baz danych oprócz pliku stogowego, ISAM lub B+ drzewa zapewniają możliwość tworzenia pliku z tak zwanym indeksem zgrupowanym (ang. *clustering index*). Indeks zgrupowany umożliwia uzyskanie dużej efektywności dostępu kosztem zwiększonego zapotrzebowania na pamięć. Indeks taki organizuje się zazwyczaj według atrybutu, który nie jest kluczem. Rekordy o takiej samej wartości wybranego atrybutu umieszcza się w jednym bloku dyskowym, dzięki czemu uzyskuje się bardzo dużą efektywność wyszukiwania rekordów według tego atrybutu. Każdemu blokowi dyskowemu odpowiada jeden rekord indeksu zgrupowanego. Niektóre relacyjne systemy baz danych umożliwiają tworzenie plików zgrupowanych, w których rekordy z różnych relacji o tych samych wartościach wybranego atrybutu są umieszczane w jednym bloku dyskowym. Organizacja taka umożliwia bardzo efektywne wykonywanie relacyjnej operacji połączenia.

W literaturze dotyczącej komercyjnych baz danych można spotkać przeciwstawienie plików typu ISAM plikom z indeksem w postaci B lub B+ drzewa. Przeciwstawienie takie może prowadzić do nieporozumień, bowiem plik z indeksem w postaci B lub B+ drzewa jest również plikiem indeksowo-sekwencyjnym. Przeciwstawienie takie ma jednak na celu uwypuklenie sztywnej organizacji indeksu i możliwości degradacji efektywności pliku ISAM wobec dynamicznej organizacji pliku typu B lub B+ drzewo.

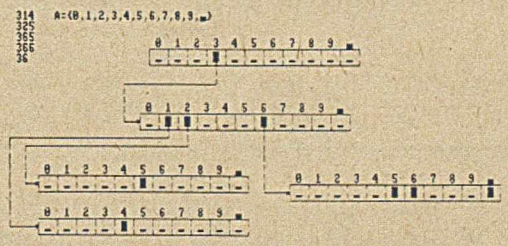
TRIE DRZEWA

Istnieje grupa drzew silnie rozgałęzionych, stosowanych przy organizacji dostępu do danych znajdujących się w pamięci operacyjnej. Do grupy tej należą tak zwane 2–3 drzewa [1, 9], będące szczególnym przypadkiem B drzew o stopniu równym 3, oraz trie drzewa [7, 8, 9].



Rys. 14. Trie drzewo

Nazwa trie drzew stanowi środkową część wyrazu *retrieval*, co odzwierciedla główne zastosowanie tych drzew jedynie do odzyskiwania informacji ze zbiorów informacji o charakterze statycznym. Trie drzewa nie są więc strukturą umożliwiającą utrzymywanie dynamicznych zbiorów danych, w których dokonuje się operacji wstawiania i usuwania. Konstrukcja trie drzew jest oparta na spostrzeżeniu, że klucz może być interpretowany jako ciąg znaków nad skończonym alfabetem. Wyszukiwanie odbywa się nie przez wielokrotne porównywanie klucza, będącego argumentem poszukiwań, z wartościami kluczy znajdującymi się w węzłach drzewa, lecz przez kolejne porównywanie znaków składowych klucza. Identyczne początkowe podciągi kluczy są w trie drzewie połączone. Ze względu na to, że ciąg odpowiadający jednemu kluczowi może być jednocześnie podciągiem innego klucza, istnieje potrzeba rozszerzenia alfabetu o dodatkowy znak oznaczający koniec klucza (rys. 14).



Rys. 15. Przykładowa realizacja struktury trie drzewa

Wyszukiwanie w trie drzewach jest bardziej efektywne niż wyszukiwanie binarne, bowiem stopień trie drzewa jest zazwyczaj większy niż dwa i równy liczbie znaków alfabetu. Czas wyszukiwania w trie drzewie jest rzędu $0(\log_a n)$, gdzie a jest przyjętą liczbą znaków alfabetu. Uzyskanie tak dużej efektywności jest jednak możliwe jedynie pod warunkiem odpowiedniej realizacji operacji porównywania znaków w węźle trie drzewa. Porównywanie to odbywać się powinno w jednym kroku, w czasie 0(1).

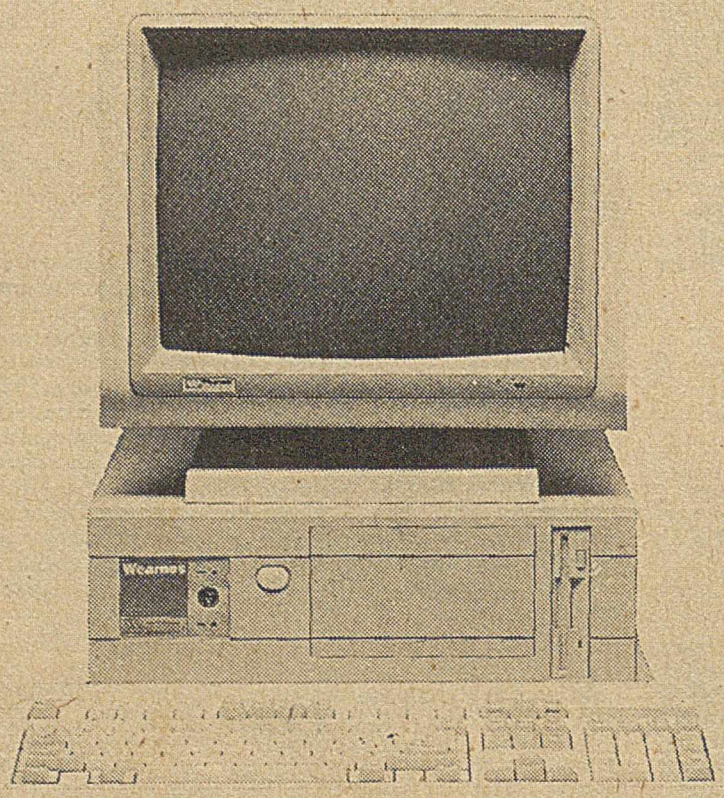
dokończenie na s. 26

KOMPUTER
Z WBUDOWANĄ PRZYSZŁOŚCIĄ

Komputer zaprojektowany przez czołowego światowego producenta mikrokomputerów – firmę ADVANCED LOGIC RESEARCH i wyprodukowany przez firmę WEARNES TECHNOLOGY.

Komputer, którego możliwości i cena oszałamiają konkurencję.

Komputer, który nie zestarzeje się, dzięki możliwości rozbudowy do architektury 386SX i 486.



Dane techniczne:

SERIA WEARNES BOLDLINE „M"

- Procesor 80286 – 12.5 Mhz
- Pamięć 1 MB RAM, możliwość rozbudowy do 16 MB (5 MB na płycie głównej)
- BIOS Phoenix
- Napęd dyskietek 3,5" 1,44 MB
- Zasilacz 110 Watt
- Klawiatura 101 klawiszy
- Podstawka dla koprocesora matematycznego 80287
- Wbudowany sterownik dyskowy z przeplotem 1:1
- Port szeregowy i równoległy
- 40 MB dysk sztywny
- Możliwość korzystania z EMS 4.0
- Obudowa typu „compact"
- Miejsce na dwa napędy 5,25" o wysokości 1/2
- Podręcznik i dyskietka z programem konfiguracyjnym
- Opcjonalna rozbudowa do 386SX i 486
- 12-miesięczna gwarancja

BOLDLINE COMPUTERS
GRUPA MICOMP-TECH
Biuro Informacji Techniczno-Handlowej
ul. Astrów 7, 40–045 Katowice
telefon i telefaks: 518-628
teleks: 315687 COMP PL

DYSTRYBUTORZY:
PTH „TECHMEX"
43–300 Bielsko-Biała
ul. M. Curie Skłodowskiej 13
tel.: 42-198, 47-555, telefaks: 47-624,
teleks: 25325

EO/1123/89

MARIUSZ POSTÓŁ
Zakład Badań Podstawowych Elektrotechniki MP i PAN
Instytut Elektrotechniki
Warszawa

Przemysłowe systemy czasu rzeczywistego (1)

Rola i funkcje oprogramowania systemowego

W pierwszej części artykułu przedstawiono wybrane wyniki analizy zagadnień związanych z konstrukcją uniwersalnego, niezawodnego oprogramowania systemowego dla przemysłowych systemów czasu rzeczywistego. Zaproponowano, aby zamiast tradycyjnego systemu operacyjnego zastosować autonomiczną bibliotekę modułów języka Modula-2. Strukturę i funkcje biblioteki wraz z przykładem praktycznego zastosowania opisano w części drugiej artykułu.

Jednym z ważniejszych obszarów zastosowań komputerów jest automatyka przemysłowa.

Terminu *komputerowe systemy przemysłowe* używa się wówczas, gdy chce się podkreślić następujące, charakterystyczne dla rozważanej klasy systemów cechy:

- konieczność bezpośredniej współpracy z urządzeniami i reagowania na zachodzące w nich zdarzenia w określonym czasie,
- niezmienność w czasie realizowanych funkcji, wynikająca z zespolenia komputera i obiektu w funkcjonalną całość,
- konieczność testowania z ograniczonym dostępem do obiektu.
- duże wymagania niezawodnościowe,

Powyższe cechy wyróżniono z uwagi na to, że najsilniej oddziaływają one na sposób projektowania i konstrukcję oprogramowania.

Systemy przemysłowe najczęściej wymagają oprogramowania działającego w czasie rzeczywistym. Zgodnie z definicją podaną przez Wirtha [17] oprogramowanie czasu rzeczywistego wyróżnia się tym, że jego poprawność zależy także od szybkości procesora lub procesorów, które je realizują. Jak wynika z praktyki, konieczność uwzględniania czynnika czasu komplikuje znacznie problem i powoduje, że omawiane systemy są powszechnie uważane za najtrudniejsze do zaprojektowania i uruchomienia.

Wraz z rozwojem sprzętu nie postępuje rozwój metod wytwarzania oprogramowania użytkowego i wspomagającego. Ta dysproporcja jest dotkliwie odczuwalna zwłaszcza przy konstruowaniu systemów należących do rozważanej klasy zastosowań, w których wymagania niezawodnościowe, realizacja oprogramowania działającego w czasie rzeczywistym i potrzeba biegłej znajomości zjawisk zachodzących w obiekcie oraz metod sterowania nimi wymagają od projektantów interdyscyplinarnej wiedzy.

Jedną z dróg, ułatwiających konstruowanie takich systemów, jest rozwój metod wspomagania wytwarzania oprogramowania działającego w czasie rzeczywistym. Celem powinno być otrzymanie efektywnych i prostych rozwiązań uniwersalnych, możliwych do wdrożenia w praktyce. Wymaga to ciągłego systematyzowania dotychczasowych osiągnięć oraz budowania na ich podstawie oprogramowania wspomagającego i systemowego.

Przykładem stworzenia uniwersalnego oprogramowania systemowego jest biblioteka DIALOGUE. Powstała ona w wyniku wieloletnich doświadczeń i analizy zagadnień związanych ze stosowaniem komputerów w przemysłowych systemach czasu rzeczywistego.

Przedstawione dalej rezultaty mogą być również wykorzystane przy konstruowaniu systemów komputerowych używanych w innych zastosowaniach, np. w medycynie, kosmonautyce, telekomunikacji, uniwersalnych systemach obsługi eksperymentu itp.

STRUKTURA OPROGRAMOWANIA

W programie systemu komputerowego wyróżnia się część systemową, zwaną systemem operacyjnym i część użytkową. **Część użytkowa** programu realizuje zadania bezpośrednio związane z konkretnym zastosowaniem. **Część systemowa** zależy od konstrukcji sprzętu i jego konfiguracji. Pozwala to traktować sprzęt i system operacyjny jako jedną całość (pewną maszynę wirtualną), która może być wyprodukowana i rozprowadzana przez jednego producenta.

Wykorzystanie modelu maszyny wirtualnej jako wzorca do budowy przemysłowych systemów czasu rzeczywistego jest może nie tyle niemożliwe [12], co w gruncie rzeczy szkodliwe. Pełna analiza tego zagadnienia wykracza poza ramy artykułu, warto jednak rozważyć wybrane aspekty tej sprawy.

Warunkiem pełnej specyfikacji maszyny wirtualnej jest określenie a priori zbioru możliwych urządzeń wchodzących w skład samego komputera i bezpośrednio w nim współpracujących. Niezbędne jest również określenie protokołów współpracy tych urządzeń z komputerem. Takie wymagania są nie do przyjęcia w przypadku komputerów przeznaczonych do kontroli bliżej nie określonych obiektów technologicznych w przemyśle. W większości zastosowań sposób współpracy i funkcje, które muszą być realizowane przez komputer, są silnie związane z charakterystyką danego obiektu technologicznego i muszą być uwzględnione w części użytkowej programu. Jak widać, granica między częścią systemową a użytkową w tej klasie zastosowań rozmywa się i często jest bardzo trudna do określenia. Stosowanie klasycznego systemu operacyjnego zmusza w związku z tym do stosowania bardzo skomplikowanych metod współpracy tych dwóch części programu; bardzo zawiła staje się również struktura całego programu.

Zgodnie z ogólnie akceptowanymi zasadami programowania strukturalnego, program powinien stanowić koncepcyjną całość utworzoną z połączonych „prefabrykatów" (modułów) o ściśle zdefiniowanych krawędziach łączących. Oczywiście poszczególne moduły mogą być projektowane i dostarczane przez różnych producentów. Program wynikowy otrzymuje się w wyniku łączenia, z tym jednak, że w razie braku systemu operacyjnego program ten musi być całkowicie autonomiczny. W rozwiązaniu takim rolę systemu operacyjnego przejmuje zbiór modułów tworzących bibliotekę systemową. Dodatkową zaletą jest to, że konkretna konfiguracja modułów jest ustalana samoczynnie podczas łączenia programu i może się zmieniać w zależności od potrzeb. Taka metodyka uwalnia od konieczności stosowania „nakładek" na i „podkładek" pod system operacyjny, czyni cały program bardziej czytelnym, a w konsekwencji ułatwia projektowanie i konserwację.

Wykorzystanie uniwersalnej biblioteki modułów jest typowym przykładem pracy w zespole, którego część stanowią projektanci modułów systemowych. Jak wiadomo [10], rezultaty pracy w zespole będą zadowalające, jeśli zostanie spełnione jedno z podstawowych zaleceń inżynierii oprogramowania, które wymaga zastosowania do formalnego zapisu programu strukturalnego języka wysokiego poziomu z wbudowanymi mechanizmami kompilacji rozłącznej. Językami spełniającymi ten warunek są m.in.: Modula-2 [15], Modula-3 [4], Ada [11], Mesa [9] lub Oberon [16].

WYBÓR JĘZYKA

Najważniejszą cechą komputerowych systemów przemysłowych jest ich niezawodność. Przy obecnym poziomie technologii wytwarzania

podzespołów elektronicznych niezawodność sprzętu komputerowego coraz częściej przewyższa niezawodność realizowanych na nim programów. Sytuacja jest nietypowa, ponieważ bardzo szybko zwiększają się zasoby stosowanych komputerów i ich moc obliczeniowa, przy praktycznie niezmiennej ich niezawodności. Pozwala to na równie szybkie rozrastanie się realizowanych przez nie programów, czego konsekwencją jest niestety najczęściej zmniejszanie się ich niezawodności.

Na zwiększenie niezawodności oprogramowania można wpływać począwszy od etapu definiowania problemu, aż do realizacji programu wynikowego włącznie. Osiąga się to przez dobrą organizację pracy w zespole projektantów, odpowiedni wybór języka programowania i użytych algorytmów oraz właściwy dobór oprogramowania wspomagającego i systemowego. Niezbędne jest również wbudowanie do programu elementów samotestowania.

Większość obiektów przemysłowych można traktować jako pewien proces technologiczny, złożony ze współbieżnie zachodzących w nim zjawisk i zdarzeń. Istotne jest to, że zawsze można wyróżnić zdarzenia, które zachodzą niezależnie i asynchronicznie względem siebie. Język użyty do zapisu algorytmu musi umożliwiać konstruowanie programu w postaci zbioru współbieżnych operacji-procesów, to znaczy takich, których wzajemna kolejność realizacji jest z góry nieokreślona, i pozwala na proste formułowanie operacji wzajemnego szeregowania (synchronizacji) procesów i wymiany danych między nimi (komunikacji).

Komunikację między obiektem a komputerem, polegającą na przepływie niezbędnych informacji o stanie obiektu, zapewnia się dwoma drogami. Zdarzenia w obiekcie, podobnie jak w komputerze, są sygnalizowane za pośrednictwem przerwań. Stan zjawisk w obiekcie jest określany natomiast na podstawie pomiarów realizowanych przez odpowiednio wyposażone kanały komputera, cyklicznie lub sporadycznie jako reakcja na sygnalizowane zdarzenia. Język musi zatem udostępniać mechanizmy lokalizowania zmiennych pod wskazanymi adresami i (lub) mechanizmy dostępu do rejestrów urządzeń we-wy oraz pozwalać na proste formułowanie operacji synchronizacji procesów programowych z procesami zachodzącymi w otoczeniu komputera za pośrednictwem przerwań.

Przy realizacji systemu przemysłowego można stosować inny komputer do wspomagania prac projektowych i inny do kontroli obiektu przemysłowego. W takim razie musi istnieć grupa programów wspomagających, w tym kompilatorów, umożliwiających łatwe przenoszenie programów użytkowych z jednego komputera na drugi i uruchamianie oraz testowanie ich na komputerze docelowym.

Ponadto, niezależnie od zastosowania, język programowania powinien:
- skłaniać (a nawet zmuszać) do programowania strukturalnego i wymagać dużej dyscypliny programowania (pełna kontrola zgodności typów), zwłaszcza przy programowaniu współbieżnym,
- zapewniać dużą czytelność programów,
- być uniwersalnym i umożliwiać łatwą implementację wielu różnych algorytmów i struktur danych,
- umożliwiać podział programu na rozłącznie kompilowane moduły i sprawdzać zgodność typów obiektów zewnętrznych dokładnie tak, jak dla obiektów lokalnych.

Wydaje się, że wobec przedstawionych kryteriów, jednym z najodpowiedniejszych języków jest obecnie kolejna propozycja Wirtha – język Modula-2 [2, 15]. Język ten ma wszystkie zalety Pascala, a nie ma jego głównych wad. W języku tym opracowano bibliotekę systemową DIALOGUE. Języki Oberon i Modula-3 nie mogły być brane pod uwagę, ponieważ ich definicje zostały opublikowane niedawno. Ada i Chill [7] mają natomiast wbudowane dość trudne do zaimplementowania mechanizmy komunikacji procesów [1]. Powoduje to, że trudno jest zrealizować w praktyce postulat dotyczący rezygnacji z systemu operacyjnego. Ponadto w okresie, kiedy dokonywano wyboru języka do implementacji biblioteki DIALOGUE, nie były dostępne kompilatory tych języków dla komputerów przemysłowych (wbudowanych).

MECHANIZMY PROGRAMOWANIA WSPÓŁBIEŻNEGO

Wyróżniającą właściwością systemu czasu rzeczywistego jest konieczność konstruowania programu w postaci współbieżnych zadań-procesów. Biblioteka systemowa powinna zatem dostarczać odpowiednie operacje umożliwiające definiowanie, inicjowanie, synchronizowanie i komunikowanie się procesów współbieżnych wykonywanych w jednym procesorze. W tej dziedzinie bardzo istotny jest również język programowania, którego elementy muszą uzupełniać i współdziałać z mechanizmami dostarczanymi przez bibliotekę.

Koncepcja programowania współbieżnego, przyjęta przez twórcę Moduli-2, jest taka, że język powinien zawierać minimalny, lecz funkcjonalnie pełny zestaw środków, umożliwiających realizację strukturalnych, wygodnych dla użytkownika operacji. Rozwiązanie takie jest uniwersalne i pozwala wykorzystać różne koncepcje i algorytmy synchronizacji, komunikacji i szeregowania procesów. Jego wadą jest natomiast brak rzeczywistej kontroli poprawności wykorzystywania dostępnych środków. Dlatego, aby zapewnić odpowiednią niezawodność programu, obszar ich stosowania powinien być możliwie mały. W bibliotece DIALOGUE ograniczono go do jednego modułu programowego, który eksportuje wygodne w użyciu operacje, spełniające wymagania programowania strukturalnego.

Przy tworzeniu programu współbieżnego najważniejszym problemem jest synchronizacja procesów w dostępie do dzielonych zasobów oraz unikanie zakleszczania. Trzeba podkreślić, że błędy programowania współbieżnego należą do tych, które najtrudniej jest wykryć i zlokalizować, niezależnie od zastosowanych metod weryfikacji poprawności programu.

Obecnie znanych jest wiele metod synchronizacji i komunikacji procesów [3, 6]. Jedną z nich jest koncepcja monitorów [5], którą wykorzystał – między innymi – Wirth w definicji języka Modula [13, 14]. Próbę tę uznał jednak za nieudaną i w następnej wersji języka – w Moduli-2 – zmodyfikował ją. Zaproponował przeniesienie wybranych mechanizmów programowania współbieżnego z samego języka do biblioteki standardowej [15]. Między tymi rozwiązaniami występuje jednak bardzo istotna różnica. W języku Modula wprowadzono jednolity mechanizm synchronizacji wzajemnej procesów oraz procesów ze zdarzeniami, sygnalizowanymi przerwaniami, zachodzącymi w otoczeniu komputera. Każdy proces może zatrzymać się (operacja WAIT lub DOIO) do chwili otrzymania oczekiwanego sygnału (wysłanego operacją SEND) lub wystąpienia zdarzenia zewnętrznego (sygnalizowanego przerwaniem). W bibliotece języka Modula-2 zrezygnowano natomiast z umieszczenia operacji synchronizacji z otoczeniem. Należy przypuszczać, że u podstaw takiej decyzji leżały dwa ważkie argumenty. W systemach komputerowych ogólnego stosowania użytkownik nie ma potrzeby bezpośredniej współpracy z otoczeniem komputera, więc nie trzeba mu udostępniać operacji synchronizacji przerwaniem. Realizując koncepcję monitorów z wbudowanym mechanizmem obsługi przerwań w języku Modula, autor języka musiał dokonać pewnych jej modyfikacji. Wyróżnił dwa rodzaje procesów: regularne (nie wykorzystujące przerwań) i urządzeń (wykorzystujące przerwania do synchronizacji ze zdarzeniami zewnętrznymi); zabronił wzajemnej synchronizacji procesów urządzeń, zmodyfikował zasadę szeregowania po operacji SEND dla procesów urządzeń itd. Rozwiązanie to nie w pełni pokrywa się z ideą Hoare'a, co chyba ostatecznie przesądziło o wyeliminowaniu procedury DOIO ze zbioru dostępnych operacji programowania współbieżnego w bibliotece języka Modula-2.

W systemach czasu rzeczywistego użytkownik często musi korzystać z mechanizmów synchronizacji za pośrednictwem przerwań, z tym że obszar jest ograniczony do procesów obsługi zdarzeń zewnętrznych, a więc jest relatywnie mały. Dyskutowana koncepcja programowania współbieżnego jest bardzo prosta w implementacji i bardzo efektywna w użyciu. Jest to istotna cecha w systemach czasu rzeczywistego, w których czas reakcji jest jednym z krytycznych parametrów. Powyższe argumenty oraz analiza porównawcza metod programowania współbieżnego [1] zdecydowały o wykorzystaniu monitorów do konstrukcji biblioteki DIALOGUE.

ZADANIA I STRUKTURA PRZEMYSŁOWYCH SYSTEMÓW CZASU RZECZYWISTEGO

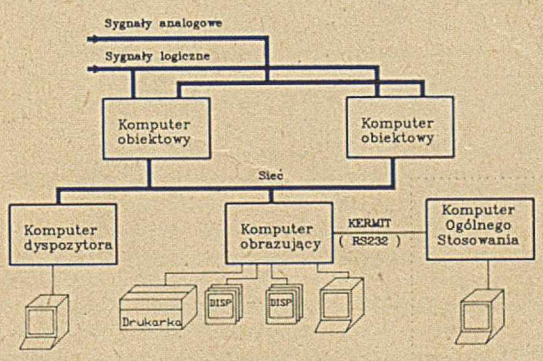
Rolą systemu komputerowego w rozważanych zastosowaniach jest przetwarzanie informacji opisującej zachowanie się kontrolowanego przez niego obiektu. Przykładami obiektów są: urządzenia technologiczne, aparatura kontrolno-pomiarowa, cała linia technologiczna, urządzenia telekomunikacyjne itp. Na ogół w tego rodzaju zastosowaniach komputer wraz z obiektem stanowi integralną całość. Fakt użycia komputera powinien być właściwie nieistotny dla użytkownika i dlatego często nazywa się je komputerami wbudowanymi.

Do typowych zadań komputerowych systemów czasu rzeczywistego stosowanych w przemyśle zaliczyć należy:

- zbieranie danych w celu odtwarzania bieżącego stanu procesów w obiekcie,
- sterowanie obiektem, tzn. dostarczanie sygnałów sterujących, w celu zapewnienia przebiegu procesów w obiekcie, zgodnego z wybranymi kryteriami jakości,
- sporządzanie dokumentacji, obrazowanie danych charakteryzujących przebieg procesów, sygnalizowanie przekroczeń i ocena jakości sterowania na podstawie analizy zachowania się obiektu,
- współpraca z dyspozytorem – wprowadzanie dodatkowych danych i poleceń,
- gromadzenie danych o zachowaniu się obiektu w czasie,
- przesyłanie danych do komputera oddalonego.

Jedną z podstawowych funkcji każdego komputerowego systemu czasu rzeczywistego jest przetwarzanie danych pomiarowych w celu odtwarzania przebiegu zjawisk w kontrolowanym obiekcie. W tym procesie wykorzystuje się odpowiednie zależności matematyczne opisujące te zjawiska. Ich doboru dokonuje się tak, aby w rezultacie uzyskać kompromis między dokładnością odtwarzania a czasochłonnością obliczeń. Realizacja tej funkcji może być tylko w niewielkim stopniu wspomagana przez oprogramowanie systemowe. Pomiar sygnałów i przetwarzanie danych muszą być realizowane cyklicznie co pewien stały odcinek czasowy. Przydatne tu więc będą mechanizmy pozwalające definiować okres powtórzeń dla wybranych operacji. Omawiana funkcja jest zwykle realizowana synchronicznie z operacją sterowania pracą obiektu.

Nie każdy system komputerowy może realizować funkcję automatycznego sterowania obiektem. Najczęściej spotykane przyczyny to brak dostatecznej wiedzy na temat obiektu do opracowania formalnych algorytmów realizowanych przez dostępne komputery i gwarantujących odpowiednią niezawodność oraz odporność na nieprzewidziane zdarzenia, w tym awarie. Niekiedy, dla bardzo skomplikowanych procesów, trudności mogą pojawić się już przy określaniu zbioru istotnych danych i sposobu ich takiego obrazowania, by w każdym momencie dawać dyspozytorowi tylko informacje niezbędne do podejmowania decyzji. Ich wybór i sposób obrazowania decyduje o szybkości percepcji, a więc rzutuje na szybkość podejmowania decyzji. Funkcja przetwarzania danych i ich obrazowania występuje praktycznie w każdym systemie komputerowym. Wyświetlanie danych wymaga dostępu do operacji wypisywania na dostępnych w systemie urządzeniach we-wy, jak monitory i drukarki. Ponieważ wiele procesów może realizować tę funkcję współbieżnie, każdy z nich powinien korzystać tylko z pewnego abstrakcyjnego urządzenia. Przyporządkowanie urządzeniom abstrakcyjnym urządzeń fizycznych jest ustalane poza tymi procesami. Dzięki temu łatwo można określić na bieżąco zestaw istotnych danych, które są obrazowane w danej chwili, bez ingerencji w sam proces obrazowania. Procesy tworzące tę funkcję muszą mieć dostęp do danych opisujących zachowanie się obiektu oraz do danych systemowych, jak data i czas. Dostęp do danych pomiarowych wymaga wykorzystania operacji komunikacji za pośrednictwem sieci lokalnej, gdy system jest rozproszony.



Struktura przykładowego, rozproszonego komputerowego systemu przemysłowego

W rozważanej klasie zastosowań konieczne jest wprowadzanie przez dyspozytora pewnych informacji dotyczących obiektu (które np. nie mogą być zmierzone) i poleceń dla systemu. Wymaga to uwzględnienia w algorytmie działania komputera funkcji komunikacji z dyspozytorem. W zależności od żądanej szybkości reakcji na polecenia stosuje się różne metody i urządzenia. Do najprostszych i jednocześnie najbardziej specjalistycznych należą różnego rodzaju przełączniki, których stan jest śledzony przez program użytkowy, podobnie jak zachowanie się obiektu. Jednak najbardziej uniwersalnym i dogodnym urządzeniem jest monitor graficzny z klawiaturą i myszą.

Stosując koncepcję okien i menu, często udaje się pokonać występującą w tym rozwiązaniu barierę szybkości wprowadzania.

W każdym większym systemie jest wymagana funkcja gromadzenia danych, opisujących zachowanie się kontrolowanego obiektu w czasie, i przesyłania ich do innego komputera. Powinna ona umożliwiać odtwarzanie przebiegu zdarzeń przed i w trakcie awarii w kontrolowanym procesie, dokonywanie analizy możliwych przyczyn awarii, rozliczanie osób odpowiedzialnych za obsługę maszyn oraz dokonywanie bilansów i obliczeń statystycznych, przy wykorzystaniu komputerów ogólnego stosowania.

Jedynym uzasadnionym zastosowaniem dla magnetycznych pamięci masowych w systemach przemysłowych jest długoterminowe archiwowanie danych opisujących zachowanie się kontrolowanego procesu w czasie. Duża zawodność tych pamięci, wrażliwość na pył i kurz oraz podatność na uszkodzenia w skutek nieumiejętnej obsługi sprawiają, że w zastosowaniach wymagających zwiększonej niezawodności trzeba unikać ich stosowania. Alternatywę stanowią pamięci elektroniczne z awaryjnym zasilaniem, zbudowane z użyciem układów CMOS. Jeśli komputer przemysłowy może transmitować dane do komputera ogólnego przeznaczenia, to rozmiar przechowywanych w nim danych można praktycznie ograniczyć do takiego stopnia, by relacja kosztów była również niekorzystna dla pamięci z nośnikami magnetycznymi.

W zależności od stopnia złożoności opisanych funkcji system może mieć różną strukturę. Jednym z najprostszych rozwiązań jest pojedynczy komputer, ewentualnie połączony z komputerem ogólnego przeznaczenia za pośrednictwem sprzęgu szeregowego. Komputer ten realizuje wszystkie funkcje w stopniu określonym wymaganiami stawianymi przez użytkownika. Odmienna struktura systemu powstaje w wyniku rozproszenia realizacji jego funkcji między niezależne komputery połączone ze sobą w sieć lokalną (rys. 1). Sieć jest tu elementem spinającym komputery w jeden integralny system. W pokazanej strukturze komputer obiektowy jest odpowiedzialny za pomiary, sterowanie i zbieranie danych. Komputer obrazujący przetwarza dane o obiekcie, w celu przedstawienia ich na ekranach monitorów; może ponadto realizować funkcję sterowania nadrzędnego. Komputer dyspozytora natomiast odpowiada za komunikację z osobą nadzorującą. Jak widać, najbardziej krytycznym (ze względu na niezawodność) elementem tej struktury jest komputer obiektowy. Często dubluje się go, w celu zwiększenia niezawodności pracy całego systemu.

LITERATURA

[1] Arend D., Postół M.: Realtime multiprogramming system for mine control centre. Microprocessors and Microsystems, Vol. 14, No. 1, pp. 39–46, 1990

[2] Arendt D., Postół M., Zajączkowski A.M.: Modula-2. NOT Warszawa, 1989

[3] Ben-Ari M.: Podstawy programowania współbieżnego. WNT, Warszawa, 1989

[4] Cardeli L. et al.: Modula-3 Report. DEC-System Research Centre, Palo 1988

[5] Hoare C.A.R.: Monitors – An Operating System Structuting Concept. Communication of ACM, Vol. 17, pp. 549–557, 1974

[6] Iszkowski W., Maniecki M.: Programowanie współbieżne. WNT, Warszawa 1982

[7] Jarociński M., Średniawa M.: Wprowadzenie do języka Chill. WKŁ, Warszawa 1988

[8] Kirk B.: Mosys – The Modula-2 System. Microprocessors and Microsystems, Vol. 8. No. 7, pp. 361–367, 1984

[9] Lapson D.W.: Experience with processes and monitors in Mesa. Communication of ACM, Vol. 23, No. 2, pp. 105–117, 1980

[10] Pomberger G.: Software Engineering and Modula-2. Prentice Hall, Englewood Cliffs 1986

[11] Pyle I.C.: Ada.WNT, Warszawa 1986

[12] Ready J.F.: VRTX: A Real-Time Operating System for Embedded Microprocessor Applications. IEEE Micro, August 1986

[13] Wirth N.: Design and Implementation of Modula. Software Practice and Experience, Vol. 7, pp. 67–84, 1977

[14] Wirth N.: Modula – A Language for Modular Multiprogramming. Software Practice and Experience, Vol. 7, pp. 3–35, 1977

[15] Wirth N.: Programming in Modula-2. Springer-Verlag, Berlin 1985

[16] Wirth N.: The programming language. Oberon. Software Practice and Experience, Vol. 18, pp. 671–690, 1988

[17] Wirth N.: Toward a Discipline of Real-Time Programming Communication of the ACM. Vol. 20, No. 8, pp. 577–583, 1977.

JAN RYŻKO
ZDZISŁAW WRZESZCZ
Instytut Maszyn Matematycznych
Warszawa

Optyczna pamięć dyskowa

Możliwość wykorzystania zjawisk optycznych do technicznej realizacji pamięci komputerowej brano pod uwagę niemal od zarania rozwoju informatyki. W pracy [1], w której zostały opisane realizacje optyczne we wszystkich dziedzinach informatyki, podkreślono, że lepsze własności fotonu w stosunku do elektronu dawały nadzieję na uzyskanie urządzeń o korzystniejszych parametrach. Nadzieje te jeszcze wzrosły po wynalezieniu lasera, kiedy skoncentrowana wiązka światła wydawała się być środkiem idealnie dostosowanym do zapisu, odczytu i wymazywania informacji. Praktyczna realizacja tego celu okazała się jednak bardzo trudna. Jak wspomniano w wymienionej pracy, wielkie nadzieje wiązane z pamięcią holograficzną, która według zapowiedzi z końca lat sześćdziesiątych miała przynieść konstruktorom kompleksowe rozwiązanie problemów pamięci, nie spełniły się. Jeszcze obecnie spotyka się publikacje na temat asocjacyjnych pamięci holograficznych [2, 3], lecz dotyczą one prac badawczych, a nie wdrożeń.

W innych rodzajach pamięci notowano natomiast stały postęp z zakresu podstawowych parametrów, co zwiększyło dystans w stosunku do technologii optycznej. Dopiero lata osiemdziesiąte przyniosły w tej dziedzinie trwałe osiągnięcia w postaci optycznej pamięci dyskowej. Najwcześniej opracowano pamięć stałą, co zbiegło się z wprowadzeniem w tym czasie na rynek płyt kompaktowych. Stąd przyjęło się powszechnie popularne określenie tego typu pamięci jako CD-ROM (*Compact Disk-Read Only Memory* [4]). Konsekwencją dalszych prac badawczo-rozwojowych było skonstruowanie dysków z wielokrotnym zapisem i wielokrotnym odczytem WORM (*Write Once, Read Many* [5, 6]), a ostatnio również dysków wymazywalnych [7] z wielokrotnym zapisem WMRM (*Write Many, Read Many*). Te właśnie pamięci są przedmiotem niniejszego opracowania.

Zakładając, że zarówno szczegółowe zasady działania pamięci optycznych jak i historia prac badawczych w tej dziedzinie zostały już wielokrotnie opisane w literaturze, ograniczymy się do przedstawienia zasad działania optycznej pamięci dyskowej, obejmujących organizację systemu dyskowego i problemy technologiczne nośnika, oraz do aktualnego przeglądu zastosowań i analizy rynku tej pamięci.

ORGANIZACJA SYSTEMU DYSKOWEGO

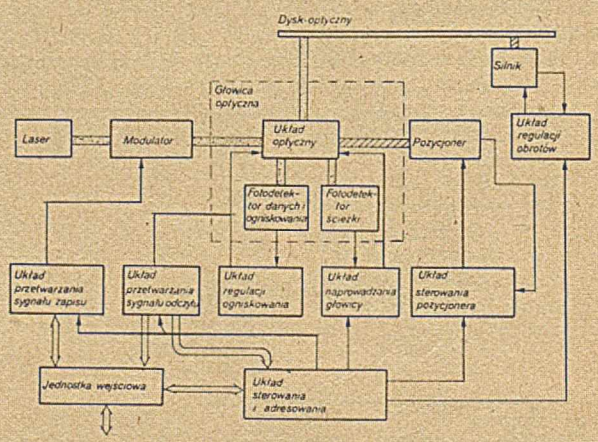
Po nieudanych próbach wprowadzenia optycznej pamięci operacyjnej (głównie holograficznej), dopiero optyczna pamięć dyskowa oparła się próbie czasu. Dziś zalicza się ją do jednego z najbardziej perspektywicznych rodzajów pamięci systemów komputerowych. Jej podstawowe zalety to duża gęstość zapisu i dobra jakość odtwarzanych informacji. Aby jednak osiągnąć obecny poziom technologii trzeba było wielu lat badań w takich dziedzinach, jak: chemia, optyka, mechanika precyzyjna, inżynieria materiałowa. Nie bez znaczenia był także równoległy rozwój podobnych dysków (płyt) do zapisu dźwięku i zapisu obrazu, co pozwoliło przyspieszyć opracowanie takich podzespołów, jak głowica laserowa wraz z układem skupiającym, układ pozycjonowania, talerz z nośnikiem, napęd i wreszcie układ odczytu informacji.

Prace nad optycznymi pamięciami dyskowymi zaczęto intensywnie rozwijać w pierwszej połowie lat siedemdziesiątych. Początkowo zapisywano informację przez wypalenie laserem otworków w warstwie metalicznej. Wiązka światła laserowego jest wówczas kierowana na dysk za pośrednictwem głowicy z układem optycznym, którą przemieszcza się w kierunku radialnym dysku do ścieżki o określonym numerze. Układ optyczny pozwala na skupienie wiązki o średnicy 1 μm (zależnie od długości fali światła), co pozwala na uzyskanie gęstości zapisu wzdłuż ścieżki rzędu 10^4bit/mm. Pojemność całej jednej strony dysku jest rzędu 10^{10}–10^{11} bitów, tzn. mniej więcej rząd wielkości więcej niż można uzyskać w magnetycznych pamięciach dyskowych. Tak duże pojemności osiąga się dzięki temu, że głowica jest naprowadzana na ścieżkę za pomocą układów automatycznie regulujących jej położenie względem ścieżki, z dokładnością do ułamka mikrometra. Równie dokładnie jest regulowana wysokość lotu głowicy, dzięki czemu punkt skupienia wiązki światła zawsze przypada w odpowiednim miejscu.

W przypadku pamięci stałej na płaszczyźnie dysku znajdują się miniaturowe otworki, których głębokość jest związana z długością fali wiązki odczytowej. Przy odczycie współczynnik odbicia wiązki światła zmienia się znacznie w zależności od ukształtowania powierzchni, co powoduje, że zmieniają się również parametry wiązki odczytowej, zapewniając wierne odczytanie zapisanej informacji.

W przypadku dysków typu WORM zapis odbywa się każdorazowo na wstępnie uformowanych otworach, natomiast w dyskach wymazywalnych (WMRN) zastosowano specjalną warstwę czynną z materiału o specjalnych własnościach magneto-optycznych bądź amorficznych. W pierwszym przypadku (magnetooptyki) następuje przy zapisie lokalna zmiana polaryzacji materiału magnetycznego, co przy odczycie powoduje zmianę położenia wektora polaryzacji wiązki. Natomiast w drugim przypadku przy zapisie występuje zmiana struktury materiału dysku – z amorficznej na krystaliczną, co oddziaływuje na dyspersję wiązki laserowej. Zapewnia to poprawny odczyt informacji przy jednoczesnym lub późniejszym wymazaniu tej informacji w celu stworzenia możliwości ponownego zapisu.

Odległość głowicy od dysku wynosi około 2 mm, dzięki czemu unika się problemów spotykanych w pamięciach magnetycznych, gdzie odległość ta wynosi zwykle ułamek μm, powodując duże kłopoty konstrukcyjne i eksploatacyjne. Nośnik jest chroniony przed uszkodzeniem za pomocą specjalnej warstwy lub przez umieszczenie dysku w kasecie. Czas dostępu jest na razie dłuższy niż w pamięciach magnetycznych.



Rys. 1. Schemat blokowy optycznej pamięci dyskowej [8]

Uproszczony schemat blokowy optycznej pamięci dyskowej pokazano na rys. 1 [8]. Dotyczy on pamięci typu WORM z jednym źródłem światła. Podstawowym zespołem jest głowica optyczna, która służy zarówno do zapisu jak i odczytu informacji. Ponadto znajduje się w niej układ optyczny do skupiania i kierowania wiązki na dysk oraz do detekcji wiązki odbitej, a także elementy sygnalizowania błędów ogniskowania lub położenia głowicy w stosunku do ścieżki informacyjnej. Odległość głowicy od dysku jest również regulowana automatycznie za pomocą układu regulacji ogniskowania.

Przy zapisie informacji wiązka laserowa jest modulowana za pomocą modulatora elektrooptycznego lub akustooptycznego sterowanego sygnałami odpowiadającymi informacji przekazywanej z komputera. W przypadku stosowania lasera półprzewodnikowego sygnały te dochodzą do układu sterującego laserem. Następnie wiązka przechodzi do układu optycznego głowicy, który ją poszerza, a następnie przez obiektyw skupiający kieruje na dysk.

Przy odczycie informacji na dysk podaje się ciągłą wiązkę laserową o mniejszym natężeniu światła. Po odbiciu od dysku wiązka ma natężenie światła zależne od zapisanej informacji. Dzięki wtrąconej w drogę przebiegu promienia płytce ćwierćfalowej i płytce światłodzielącej wiązka odbita od dysku jest kierowana do detektora światła. Stamtąd sygnał odczytu przechodzi do układu przetwarzania, skąd przez jednostkę wejściową pamięci jest przekazywany do komputera.

Głowica powinna być utrzymywana w stałej odległości od dysku z dokładnością ± 0,5 μm, ponieważ zwykle tyle wynosi głębia ogniskowania. Układ regulacji prędkości zapewnia stałą prędkość obrotową dysku, co powoduje stałą prędkość transmisji danych. Dysk obraca się zwykle z prędkością ograniczoną do 30 obr/s.

PROBLEMY TECHNOLOGICZNE NOŚNIKA (DYSKU)

Spośród wszystkich podzespołów występujących w dyskowej pamięci optycznej największe problemy nastręcza technologia samego nośnika informacji. Dlatego tym problemem zajmiemy się bardziej szczegółowo, opierając się częściowo na prospektach firmy Daicel Chemical Industries, Ltd [9], będącej doświadczonym producentem dysków typu WORM i przygotowanej do produkcji dysków wymazywalnych.

Ośrodek pamięciowy powinien charakteryzować się:
- dużą czułością,
- dużą rozdzielczością,
- dużym stosunkiem sygnału do szumu,
- brakiem defektów i odpornością na zanieczyszczenia,
- dużą trwałością zapisu.

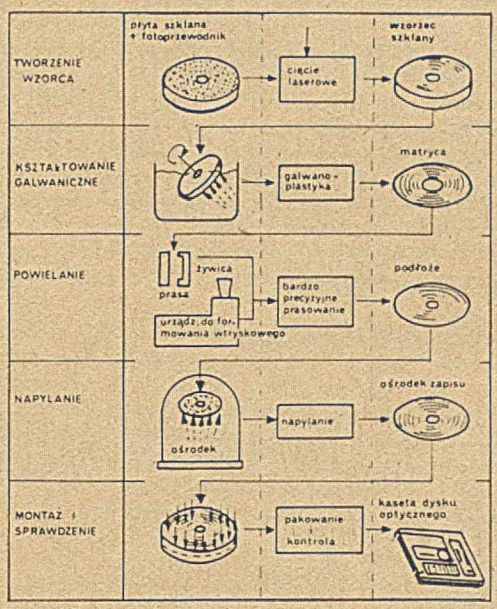
Nośnikom wymazywalnym stawia się dodatkowe wymagania:
- wymazywanie i zapis powinny przebiegać z szybkością rzędu 1 Mbajta/s,
- wymazywanie powinno odbywać się bit po bicie i być odróżnione od zapisu,
- warstwa lub położenie nie mogą ulegać deformacjom pod wpływem promieniowania.

Przy zapisie trwałym (WORM) metoda zapisu informacji polega na usunięciu warstwy metalicznej przez punktowe nagrzanie za pomocą wiązki laserowej lub wybrzuszenie tej warstwy. W pewnych rozwiązaniach [10] stosowano tworzenie otworów w warstwie metalicznej o grubości 30–50 nm, naniesionej próżniowo na podłoże ze szkła lub PMMA (polimetakrylen metylu). Ośrodek taki odbija 40% energii, pochłania 40%, a przepuszcza 20%. Lepsze rezultaty uzyskano stosując strukturę trójwarstwową [11], gdzie podłoże pokrywa się warstwą aluminiową odbijającą światło, na którą nanosi się warstę izolacyjną przepuszczającą światło, następnie nakłada się bardzo cienką warstwę metaliczną. Warstwa metaliczna i dielektryczna, przez odpowiednio dobrane grubości (wielokrotność $\lambda/4$), tworzą strukturę antyrefleksyjną. Dzięki temu prawie 90% światła ulega absorbcji przez warstwę metaliczną. Materiałem metalicznym jest najczęściej tellur. Aby warstwa aluminiowa nie odprowadzała ciepła wydzielonego w warstwie metalicznej, stosuje się dielektryk polimerowy o dużej odporności termicznej, zamiast używanego zwykle SiO_2.

Przy wybrzuszeniach metaliczna warstwa absorbująca ma wysoką temperaturę topnienia (np. tytan), a dielektryk niską temperaturę parowania. Podgrzewając laserem warstwę metaliczną powoduje się parowanie dielektryka w pobliżu miejsca naświetlania. Ciśnienie gazów rozciąga warstwę metaliczną – przy odczycie występuje tu odbicie wiązki, natomiast w miejscach niezdeformowanych – jej absorbcja.

Jak wspomniano poprzednio, do struktur wymazywalnych stosuje się materiały magnetooptyczne, lub krystaliczno-amorficzne. Są to zwykle struktury czterowarstwowe. Na podłożu znajduje się aluminium a następnie, między dwoma warstwami dielektryka, ów materiał wymazywalny. Wymazywanie informacji odbywa się przez naświetlanie warstwy światłem o mocy zbliżonej do zapisu, lecz trwającym znacznie dłużej. Odczyt odbywa się wiązką małej mocy. W pamięciach produkowanych przez firmę Matsushita [12] materiałem jest podtlenek telluru z domieszką germanu, indu i ołowiu. Również w ostatnich doniesieniach [13] o urządzeniach wprowadzanych do produkcji, mowa jest o nośniku amorficznym.

Przy magnetooptycznym typie materiału wiązka światła przy zapisie podgrzewa miejscowo warstwę magnetyczną wskutek czego staje się ona podatna na zmianę kierunku magnesowania pod wpływem zewnętrznego pola magnetycznego. W ten sposób w miejscach naświetlonych tworzą się domeny o kierunku namagnesowania przeciwnym do wytworzonego uprzednio dla całego dysku. Podczas odczytu, dokonanego światłem o mniejszej energii niż przy zapisie, wykorzystuje się efekt Kerra – skręcania płaszczyzny polaryzacji światła. Kierunek skrętu zależy od kierunku namagnesowania warstwy, co pozwala na rozróżnienie zapisanej informacji. Jako warstwę magnetyczną stosuje się Gd-Co, Gd-Fe, Tb-Fe.



Rys. 2. Proces wytwarzania dysku optycznego [9]

Przy tego typu materiałach można wyróżnić [9] pięć etapów procesu wytwarzania dysku optycznego. Są one pokazane na rys. 2. Dokładnie wypolerowana płyta szklana pokryta jest fotooporniki em, którym jest fotoczuły polimer. Na płytę tę są nanoszone rowki ścieżek oraz wstępne formatowanie za pomocą wiązki światła z lasera argonowego. Na ten szklany wzorzec nanosi się cienką warstwę przewodzącą, a następnie metodą galwanoplastyczną tworzy się niklową matrycę, którą umieszcza się w urządzeniu do formowania wtryskowego. Urządzenie to umożliwia wytwarzanie tysięcy plastikowych podłoży, będących duplikatem szklanego wzorca. Na każdym podłożu tworzy się w procesie napylania warstwę zapisującą. Kilka podłoży z warstwą zapisu łączy się razem i pakuje w kasetę.

Ponieważ dla otrzymania zapisu o dużej gęstości wykorzystuje się obszary o rozmiarach mniejszych od 1 μm, to nawet małe uszkodzenia spowodowane w którymkolwiek z opisanych procesów będą powodować przy kontroli odrzucenie dysku jako wadliwego. Dlatego ogromna precyzja i czystość są konieczne w każdym etapie procesu produkcyjnego. Uszkodzenie dysku wyraża się współczynnikiem BER (*Bit Error Rate* – częstość błędów bitowych) lub RED (*Raw Error Density* – gęstość dużych uszkodzeń).

W materiałach [9] szczegółowo opisano urządzenie do tworzenia rowków za pomocą wiązki laserowej. W urządzeniach tych lasery He-Cd są stosowane dla dysków kompaktowych, natomiast argonowe – dla wymazywalnych dysków magnetooptycznych. Celem uzyskania pojemności rzędu 400 MB, odległość ścieżek powinna być rzędu 1,6 μm, co wymaga bardzo precyzyjnego tworzenia rowków, zwykle za pomocą podwójnej wiązki laserowej.

Wymagana jest także duża jednorodność grubości matrycy (rzędu 1%), co zmusza do bardzo dokładnego kontrolowania procesu galwanoplastycznego, a następnie polerowania powierzchni matrycy. Urzą-

dzenia do wtryskowego formowania muszą być również bardzo precyzyjne, zapobiegające tworzeniu się pęcherzyków w płynnej żywicy, co wymaga stosowania odpowiedniej temperatury, szybkości chłodzenia i ciśnienia prasowania. Czynniki te określają bowiem mechaniczne i optyczne własności podłoża.

Szczególne wymagania narzuca proces napylania, który wpływa na podstawowe parametry warstwy zapisu (siła koercji, kąt skręcenia płaszczyzny polaryzacji). Wreszcie bardzo istotny jest proces montażu i kontroli. Kontrola obejmuje takie parametry, jak współczynniki odbicia (z dokładnością 5%) i przepuszczania (10%), stosunek sygnału do szumu (70 ± 1,5 dB), przesłuch (60 ± 1,5 dB), BER (w zakresie 10^{-1} do 10^{-8}) oraz defekty o wielkości od 0,6 do 75 μm w siedmiu zakresach. Przy badaniach tych jest stosowana specjalna aparatura komputerowa z oprogramowaniem kontrolnym. Badana jest również niezawodność dysków w trakcie eksploatacji w różnych warunkach otoczenia (temperatura, wilgotność, zanieczyszczenia atmosfery, wibracje, udary).

Według tych samych informacji prace badawcze nad magnetooptycznymi ośrodkami wymazywalnymi obejmowały początkowo związki *TbFe* i *GdTbFe*, a w ostatnim okresie *TbFeCo* i *NdDyFeCo*, które wykazywały lepsze parametry niezawodnościowe. Powinny one pozwolić na uzyskanie dysków wymazywalnych drugiej generacji o szybkości przesyłania danych przekraczającej 10 Mb/s, szybkości obrotów wynoszącej 3600 obr/min, stosunku sygnału do szumu większym od 45 dB oraz czułości zapisu mniejszej od 10 mW.

ZASTOSOWANIA I RYNEK

Skonstruowanie optycznych pamięci dyskowych o parametrach takich, jakie uzyskują ich odpowiedniki magnetyczne, okazało się zadaniem bardzo trudnym. Główna trudność to wielokrotny zapis. W związku z tym najpierw wprowadzono na rynek optyczną pamięć dyskową, wprawdzie o wielkiej pojemności (~ 1 G bit), lecz z możliwością tylko jednokrotnego zapisu (WORM). Okazało się, że rozwiązanie to znalazło wiele zastosowań, co zachęciło wiele firm do uruchomienia seryjnej produkcji tego typu pamięci (tabela 1).

Zastosowanie pamięci typu WORM

Pamięci typu WORM zastosowano w systemach gromadzenia danych, charakteryzujących się szczególnie dużą pojemnością zbiorów. Dotyczy to zwłaszcza gromadzenia różnych form obrazów, takich jak dokumentacja techniczna, zdjęcia mikroskopowe, zdjęcia lotnicze itp. W pierwszych rozwiązaniach konstrukcyjnych średnica dysku wynosiła 300 i 200 mm, później (po 1985 r.) producenci wybrali średnicę 130 mm, a więc standard stosowany dla dysków magnetycznych w komputerach osobistych [9]. Równolegle rozpoczęto też prace nad rozszerzeniem konfiguracji standardowego komputera osobistego o pamięć optyczną typu WORM. W tabeli 2 podano zestawienie ważniejszych typowych zastosowań tych pamięci. Warto podkreślić, że w wymienionych zastosowaniach ograniczony charakter funkcji zapisu nie jest tu wadą, lecz zaletą.

Zastosowanie pamięci optycznych z wielokrotnym zapisem (typu WMRM)

Od roku 1988 wzrosła znacznie na rynku zachodnim liczba producentów dyskowych pamięci z wielokrotnym zapisem (tabela 3). Są to w większości dyski o wymiarze 5 1/4 cala, dostosowane do współpracy z minikomputerami klasy PC AT, lecz o pojemności znacznie mniejszej niż dyski typu WORM. Pojemność ta jest jednak na tyle znaczna, że zachęca do zastosowania dysku optycznego jako pamięci systemu komputerowego. Pierwszym takim systemem jest NEXT [15], jeden z najnowszych modeli osobistych komputerów, którego moc obliczeniowa jest zbliżona do mocy maszyn dużych (ang. *main-frame*).

Tabela 2. Zastosowanie optycznych pamięci dyskowych typu WORM

Zastosowanie	Oprogramowanie systemu	Cechy sprzętu
Wsparcie dla systemu wielozadaniowego	DOS dla dysku WORM programy obsługi dla zbiorów WORM	jednostka sterująca WORM
System gromadzenia zapisów obrazów i grafiki	edytor graficzny edytor obrazów dBase III	jednostka sterująca kamerą video jednostka sterująca WORM
System gromadzenia zapisów mowy System informacyjny mowy	program kodowania mowy program syntezy mowy dBase III	jednostka sterująca mikrofonem i głośnikiem jednostka sterująca WORM
System grupowej obsługi zbiorów	programy obsługi sieci PC Lan	Sieć PC Lan jednostka sterująca WORM
Wsparcie dla telefaksu	edytor obrazów dBase III	telefaks jednostka sterująca WORM

Komputer NEXT, produkt firmy NEXT Inc., Palo Alto, USA, ma następującą charakterystykę:

- mikroprocesor Motorola 68030,
- pamięć główna (systemowa) i wspomagająca na dysku optycznym, wymazywalnym o pojemności 256 MB,
- magistrala systemu: zmodyfikowany NuBUS, 25 MHz,
- system operacyjny typu UNIX,
- drukarka laserowa o rozdzielczości 300 punktów na cal,
- cena systemu: ok. 8500 USD,
- termin pierwszych dostaw III kwartał 1989 r.

Tego rodzaju parametry pozwalają stosować ten komputer jako system wielodostępny o bardzo szerokich możliwościach aplikacyjnych.

Wprowadzenie dysku optycznego jako pamięci systemowej w nowym komputerze liczącej się na rynku USA firmy wskazuje, że technologia dysku optycznego osiągnęła już wysoki poziom rozwoju i skutecznie może konkurować z dyskami magnetycznymi.

Zapotrzebowanie na optyczne pamięci dyskowe

Odbiorcami dysków optycznych pamięci dyskowych z wielokrotnym zapisem są firmy produkujące sprzęt komputerowy (OEM – *Oryginal Equipment Manufacturers*). Prognozy niektórych firm (np. CAP International, Francja [16]) dotyczące sprzedaży dysków o wymiarach 3,5 oraz 5,25 cala w najbliższych latach przedstawiono w tabeli 4.

Prognozy te biorą niewątpliwie pod uwagę obecną wiekszą popularność mikrokomputerów wyposażonych w dyski o standardzie 5,25 cala.

Chłonność rynku na dyski optyczne z wielokrotnym zapisem będzie wzrastać w miarę poprawy parametru określającego czas dostępu. Nie

Tabela 1. Parametry seryjnie produkowanych optycznych pamięci dyskowych typu WORM

Producent	Nazwa modelu	Średnica dysku (cale)	Pojemność (Mb)	Czas dostępu (ms)	Rozmiary obudowy (cm)	Uwagi
ATG	6000	12,00	6400	100	60 × 40 × 18	nowe opracowanie
HITACHI	0K 101	5,25	48 × 600	1000	50 × 60 × 130	cena ok. 26000 USD
KODAK	6800	14,00	6800	100	kaseta 19"	możliwość przyłączenia do IBM 370
LASER DRIVE	300	5,25	820	175	8 × 21 × 14	
LMS	LD 510	5,25	654	75	8 × 21 × 14	bloki 64 KB
MITSUBISHI	MW 5D1	5,25	600	80	8 × 21 × 14	cena ok. 4100 USD
OPTIMEN	4000	12,00	4000	brak danych	60 × 44 × 17	cena ok. 17900 USD
PENTAY	LW 501	5,25	600	58	8 × 14 × 21	bloki 64 KB
PANASONIC	LF 500	5,25	20000 (50 krążków)	brak danych	48 × 10 × 70	
RICOH	RS 5160	5,25	20 krążków	brak danych	70 × 73 × 48	zgodne z Maxtor i Daicel
TOSHIBA	WM 5070	5,25	900/600	90	20 × 40 × 11	

Tabela 3. Parametry seryjnie produkowanych optycznych pamięci dyskowych z wielokrotnym zapisem typu WMRM

Producent	Nazwa modelu	Średnica dysku (cale)	Pojemność (MB)	Czas dostępu (ms)	Rozmiary obudowy (cm)	Uwagi
AGA	1000	5,25	650	61	15 × 25 × 40	bloki 256 KB
CANON	EC 512	5,25	512	80	14 × 8 × 21	bloki 64 KB – wybrany przez firmę NEXT
MAXELL	Disk	5,25	600	brak danych	brak danych	elektronika opracowywana przez Hitachi
RICOH	RO 5030 E	5,25	594	61	14 × 8 × 21	
SONY	SMO 501	5,25	650/594	95	14 × 8 × 20	cena ok. 4.650 USD
VERBATIM	TMO	3,5	60 (jednostronny zapis)	30	4 × 10 × 14	cena ok. 1.000 USD

ma jednak podstaw do twierdzenia, że dyski optyczne całkowicie wyeliminują dyski magnetyczne.

Zapotrzebowanie na dyski optyczne typu WORM osiągnie wkrótce pewne nasycenie. Wynika to po pierwsze z ograniczonej przydatności systemów specjalizowanych, jakimi są systemy gromadzenia informacji stałych oraz z nieuniknionego spadku ceny dysków WMRM, w miarę rozwoju technologii i skali produkcji.

Tabela 4. Przewidywana sprzedaż pamięci optycznych z wielokrotnym zapisem w latach 1989–1991

Format dysku (cale)	Lata		
	1989	1991	1992
3,5	3 000	22 000	50 000
5,25	10 000	100 000	180 000

W odniesieniu do pamięci z wielokrotnym zapisem należy zwrócić szczególną uwagę na:
- daleko posuniętą standaryzację rozwiązań,
- znaczną pojemność przy małych gabarytach,
- znaczną liczbę producentów,
- niezbyt wygórowaną cenę.

Są to zalety w porównaniu do pamięci magnetycznych typu WINCHESTER, natomiast istotną wadą jest niestety dość znaczny nadal czas dostępu. Brak jest też wiarygodnych danych na temat niezawodności pracy dysku optycznego przy zakłóceniach elektromagnetycznych.

OPTYCZNE PAMIĘCI DYSKOWE W POLSCE I KRAJACH RWPG

Prace badawcze w dziedzinie optycznych pamięci dyskowych nie są obecnie w Polsce prowadzone. W połowie lat osiemdziesiątych Zakłady Radiowe Kasprzaka podjęły współpracę z firmą Thomson CSF nad gramofonem laserowym, co mogło być podstawą do pracy nad pamięcią dyskową; prace te jednak przerwano. Również OBR Elektronicznego Sprzętu Powszechnego Użytku – Łódź (dawniej Zjednoczenie UNITRA) występował w roku 1985 z wnioskiem o zakup licencji na produkcję gramofonów laserowych firmy Philips. Wniosek ten jednak nie został zrealizowany. W początku lat osiemdziesiątych znaczne zainteresowanie tematem wykazywały także Zakłady ELWRO, co jednak skończyło się jedynie rozpoznaniem aktualnego stanu technologii.

W roku 1985 byłe Ministerstwo Hutnictwa i Przemysłu Maszynowego zleciło zespołowi ekspertów pod kierunkiem prof. B. Paszkowskiego opracowanie *„Programu rozwoju optoelektroniki dla potrzeb gospodarki narodowej do 1995 roku"*. W ramach tego przedsięwzięcia został opracowany program grupowy nr 04 pn. *„Optoelektronika komputerowa"* [14], zawierający podgrupę *„Pamięci optyczne"*. Podstawowym celem tego programu było sformułowanie pakietu zadań produkcyjnych po to, by zwiększyć atrakcyjność polskiej oferty na rynkach zagranicznych. Program *„Optoelektronika komputerowa"* nie został podjęty przez wspomniane Ministerstwo z przyczyn poza merytorycznych (kryzys?) chociaż, jak to pokazuje obecny rozwój problematyki na Zachodzie, był to program sformułowany bardzo trafnie i prawidłowo [1].

Prace nad optycznymi pamięciami dyskowymi prowadzą również: ZSRR, Węgry i Czechosłowacja w ramach programów badawczych SM EMC i JS EMC. Tempo tych prac jest jednak powolne. Np. w 1980 r. Węgry zgłosiły do planu opracowań pamięć o symbolu SM 590 (parametry: 0,5 – 1GB, 100 ms). Badania prototypu, założone na 1983 r. dotąd nie zostały zrealizowane. Podobnie przedstawia się sprawa z pamięcią radziecką EC 5150.

* * *

Opracowanie w Polsce optycznej pamięci dyskowej wymaga potraktowania problemu w sposób kompleksowy, co wymaga objęcia programem:
- wszystkich prac technologicznych dotyczących m.in. źródła światła (laser półprzewodnikowy), nośnika informacji oraz układu pozycjonującego i napędu;
- prac systemowych dotyczących układów zapisu, odczytu oraz sterowania.

Ponieważ liczne rozwiązania technologiczne i systemowe zostały w tym temacie już zastrzeżone patentami, niezbędne jest wykonanie w pierwszej kolejności starannego studium stanu patentowego poszczególnych węzłów problemu.

Należy stwierdzić, że bardziej perspektywiczne są pamięci z wielokrotnym zapisem, ze względu na możliwości pełnienia przez nie funkcji pamięci systemowej mikrokomputera. Tu należy więc kierować większość przeznaczonych na badania środków, ponieważ dalszy rozwój (ilościowy i jakościowy) systemów mikrokomputerowych jest oczywisty.

LITERATURA

[1] Wrzeszcz Z., Ryżko J., Sikorski A., Synak R.: Komputery optoelektroniczne. Prace naukowo-badawcze Instytutu Maszyn Matematycznych, nr 1, 1987
[2] Psaltis D., Hong J.: Shift invariant optical associative memories. Optical Engineering. No. 1, 1987
[3] Yu F., Zhang C.: Application of holographic associative memories to hybrid binary adder. Proc SPIE, Vol. 883, January 1988
[4] Controt F.: CD-ROM in boom servis a demain. Micro-Systemess, Juin 1988
[5] Willams T.: WORM and erasable media end drives, OIS International 1988, June 1988
[6] Rephogle R. A.: Optical storage (WORM) applications for the PC, PS/2. DIS International 1988
[7] Schoen A., Bresilon C.: Les disques optiques effacables, Micro-Systemes, Juin 1988
[8] Synak R.: Optyczne pamięci dyskowe. Techniki komputerowe, nr 1, 1986
[9] Materiały katalogowe firmy Daicel Chemical Industries, Tokio – Düsseldorf, 1988
[10] Harris A. L.: Continuous wave laser recording on metallic thin film. Image Technol: April–May 1970
[11] Bell A. E., Spony F. W.: Antireflection structures for optical recording. J. Quant. El., vol 17, 1981
[12] Erasable optical data storage. Laser Focus, No. 6, 1984
[13] Tandy announces breakthrough in erasable optical media. Computer Design, No. 10, 1988
[14] Wrzeszcz Z. i inni: Program pn. „Optoelektronika Komputerowa". Opracowanie będące częścią „Programu rozwoju optoelektroniki dla potrzeb gospodarki narodowej do 1995 roku". Kierownictwo: prof. B. Paszkowski, Ministerstwo Hutnictwa i Przemysłu Maszynowego, Warszawa 1985
[15] Mini – Micro Systems, Dec. 1988, 19–24
[16] Mini et Micro, Nr. 314, Jan. 1989, 51–52

Przypominamy, że zlecenia na umieszczenie ogłoszeń w INFORMATYCE przyjmuje w bieżącym roku bezpośrednio nasza redakcja. Czekamy na ogłoszenia.

MEDCOM

ZASILANIE AWARYJNE

ZASILACZE BEZPRZERWOWE

naszej produkcji

UPS–200
UPS–300

produkcji tajwańskiej

UPS–500

produkcji japońskiej

GENERATORY PRĄDOTWÓRCZE

różnych mocy

EO13/90

Biuro
Al. Ujazdowskie 26/39
00–478 Warszawa tel. 28 93 57, teleks 81 70 60

ZAKŁAD PRODUKCYJNO-USŁUGOWY
ul. Wołodyjowskiego 41
02–724 Warszawa tel.43 04 41

WITOLD REKUĆ
Instytut Organizacji i Zarządzania
Politechnika Wrocławska

Odwzorowania w relacyjnej bazie danych obiektów o nieokreślonych a priori typach

Problem przedstawiony w artykule dotyczy projektowania systemów informatycznych realizowanych z użyciem takiego oprogramowania, jak: dBase III Plus, Clipper i Foxbase. Większość tych systemów stosuje się w takich warunkach, w których struktury danych są stałe, przy ustalonym przedmiocie, zakresie i funkcjach systemu informatycznego. Zmiana przedmiotu, zakresu lub funkcji systemu wymaga na ogół zmian w strukturze bazy danych, procedur i form prezentacji danych, co jest kosztowne i czasochłonne. Oczywiście przy modyfikacjach systemu nakłady finansowe i czasowe są nieuniknione, ale ich wielkość zależy od stopnia przystosowania systemu do modyfikacji.

Zaprojektowany i zrealizowany przez autora system jest przeznaczony do ewidencji produkcji hutniczej o nieustabilizowanym (wręcz nieokreślonym) asortymencie [2]. Produkowane wyroby były opisane przez specyficzne im zestawy atrybutów technologicznych i strukturalnych. Zestawy tych atrybutów zostały ściśle określone dla wyrobów aktualnie produkowanych, natomiast nie można ich było określić dla wyrobów produkowanych w przyszłości. Postawiono wymaganie, by system informatyczny był minimalnie czuły na zmiany w asortymencie wyrobów.

Aby rozwiązać problem wrażliwości na zmiany asortymentu wyrobów, autor rozważył trzy podstawowe warianty (idee) jego konstrukcji, określone przede wszystkim przez strukturę bazy danych.

SFORMUŁOWANIE PROBLEMU

Można przyjąć, że każdy obiekt ze zbioru obiektów (dziedziny przedmiotowej) jest charakteryzowany przez zestaw specyficznych dla niego atrybutów. Obiekty charakteryzowane przez ten sam zestaw atrybutów stanowią zbiór obiektów jednorodnych (pewien typ). Obiekty charakteryzowane przez różne zestawy atrybutów są zbiorem obiektów niejednorodnych (różnych typów).

Niezależnie od kryterium jednorodności obiektów, w praktyce stosuje się także nieformalne kryteria grupowania obiektów (kryteria nieformalne można sprowadzić do kryteriów formalnych, jeżeli uwzględni się np. związki między obiektami). Na przykład, zbiór kontrahentów pewnej firmy charakteryzowanych przez identyczne zestawy atrybutów, można podzielić na dwa (być może rozłączne) typy: *DOSTAWCA I ODBIORCA*. Analogicznie, obiekty różnych typów łączy się w jeden zbiór niejednorodny. Na przykład, wyroby są łączone w zbiór oznaczony terminem *WYRÓB*, mimo że mogą być obiektami niejednorodnymi.

Przedmiotem omawianego systemu informatycznego niech będzie zbiór obiektów niejednorodnych, oznaczony terminem W i będący sumą teoriomnogościową rozłącznych i jednorodnych zbiorów W_1, W_2, ..., W_n. Dla każdego zbioru W_i jest możliwe określenie:
– typu T_i, czyli zestawu atrybutów obiektów należących do W_i,
– (typu) procedur przetwarzania P_i,
– (typu) form prezentacji informacji o obiektach należących do W_i na ekranie monitora lub papierze drukarki F_i.
W fazie projektowania systemu projektant nie dysponuje informacją o wszystkich typach elementów zbioru W.

Celem jest znalezienie takiej konstrukcji systemu – bazy danych, procedur i form prezentacji – która zapewni minimalną wrażliwość systemu na zmiany w zbiorze W. Wrażliwość ta jest rozumiana jako właściwość systemu określająca zakres zmian, które muszą być dokonane w strukturze systemu przy zmianie zbioru W, tj. przy rozszerzaniu go o nowe typy obiektów.

Problem ten jest szczególnie istotny, gdy system jest realizowany jako relacyjna baza danych. Koncepcja jednorodności jest fundamentalna dla modelu relacyjnego. Faktycznie relacje (pliki) są reprezentantami zbiorów obiektów jednorodnych. Dlatego przy projektowaniu odwzorowania dziedziny przedmiotowej w relacje szuka się w niej właśnie podzbiorów jednorodnych. Niejednorodność obiektów w danym zbiorze ma następujące konsekwencje dla systemu informatycznego:
- obiekty muszą być odwzorowane w relacyjnej bazie danych akceptującej jednorodność,
- obiekty niejednorodne wymagają różnych procedur przetwarzania (niejednorodność procedur),
- obiekty niejednorodne wymagają różnych form prezentacji (niejednorodność form prezentacji).

Wymaganie minimalnej wrażliwości systemu na skład zbioru W przypomina znaną cechę systemów informatycznych – uniwersalność. Tworzenia systemów uniwersalnych zwykle wywołuje kontrowersje. Systemy uniwersalne są bardziej złożone, wymagają większych nakładów pracy, bywają także mniej sprawne, tj. wolniejsze i bardziej pamięciochłonne. Z punktu widzenia użytkownika sprawność systemu w zastosowaniach, np. administracyjnych, może mieć jednak znaczenie drugoplanowe. Ważniejsza jest właśnie zdolność systemu do adaptacji, odpowiednio do zmian w dziedzinie przedmiotowej lub zmian w wymaganiach użytkownika.

Każdy system może być uniwersalny tylko w pewnych granicach. Ograniczenia uniwersalności są zwykle wynikiem kompromisu między złożonością, kosztem systemu, a jego właściwościami użytkowymi. Pośrednio mogą one także wynikać z właściwości systemu programowania, w którym system informatyczny jest realizowany.

WARIANTY ROZWIĄZANIA PROBLEMU

Prezentując warianty konstrukcji systemu, pozwalające rozwiązać problem czułości na zmianę składu zbioru W, posłużono się uproszczonym przykładem, w którym zbiór W zawiera wyroby charakteryzujące się określonymi atrybutami technologicznymi. W fazie projektowania w zbiorze W są następujące typy obiektów:

WYRÓB_1 (ILOŚĆ, WSAD, ELEKTRODY, ENERGIA)
WYRÓB_2 (ILOŚĆ, CZAS_PRACY_WYPADEK)
WYRÓB_3 (ILOŚĆ, CZAS_PRACY_SUSZAREK)

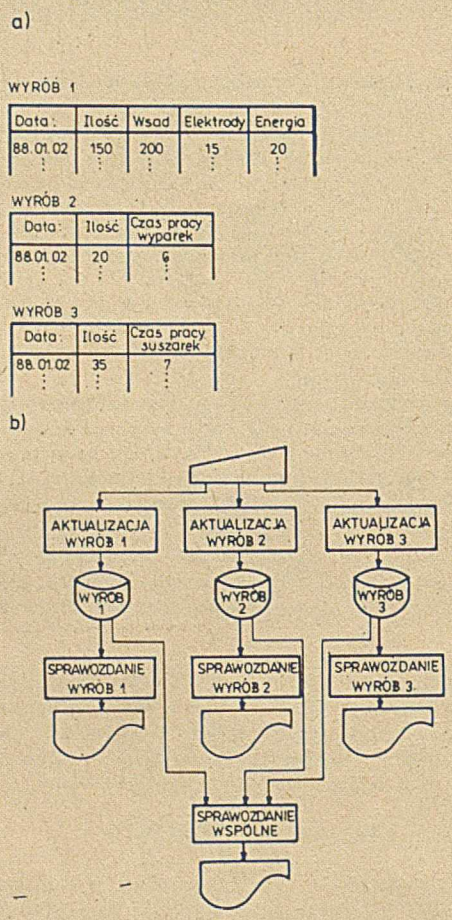
Zakłada się także, że każdy, nawet hipoteczny, wyrób ma atrybut ILOŚĆ i nie więcej niż trzy atrybuty specyficzne. Wartości wszystkich atrybutów są zmienne w czasie i mają być pamiętane wraz z datą.

Ponadto przyjęto, że w systemie zostaną zrealizowane następujące funkcje użytkowe:
1) ewidencja danych o wyrobach,
2) tworzenie sprawozdań o miesięcznej produkcji poszczególnych wyrobów,
3) tworzenie sprawozdania rocznego o wielkości produkcji globalnie dla wszystkich wyrobów.

System z niezależnymi plikami bazy danych

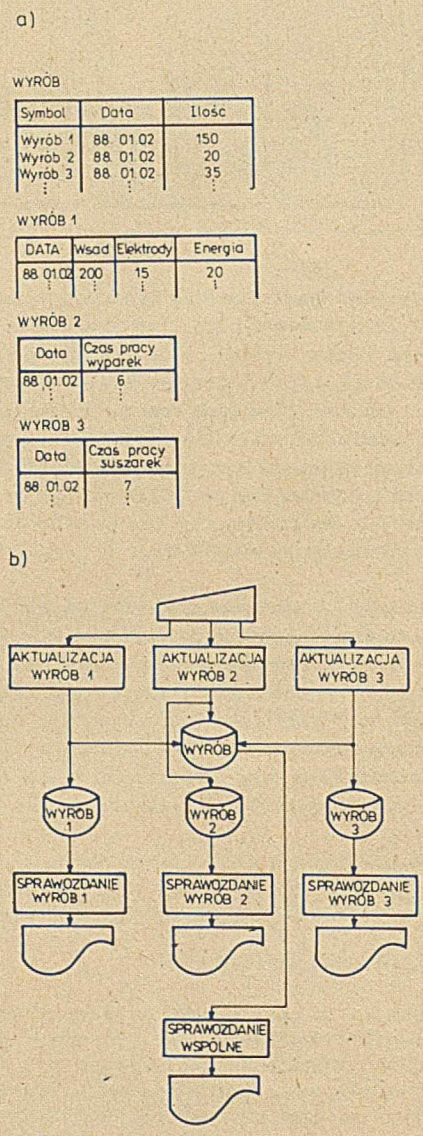
Rozwiązanie to jest stosowane powszechnie jako rozwiązanie wzorcowe. Dla wszystkich typów, które są znane w fazie projektowania systemu, projektuje się odrębne pliki (rys. 1), procedury i formy prezentacji. Ewentualnie projektuje się także procedury działające na kilku plikach, a także przewiduje się wspólne formy prezentacji. W przykładzie zrealizowanym przez autora, każdy plik był obsługiwany

przez specyficzne mu procedury aktualizacji oraz formy prezentacji na ekranie monitora. Przewidziano także jedną procedurę przetwarzającą wszystkie pliki. Istota tego wariantu polega na odrębnym opisie każdego typu obiektów. Rozwiązanie to jest proste, cechuje je duża sprawność i mały koszt realizacji. Jest ono natomiast czułe na zmianę asortymentu wyrobów (obiektów w zbiorze W). Pojawienie się nowego typu T_j wymaga zaprojektowania i zrealizowania nowego systemu, rozszerzonego o odpowiedni plik, procedury i formy prezentacji. Jeżeli w systemie są procedury działające na wszystkich plikach, to także muszą być zmodyfikowane. Realizacja funkcji użytkowych systemu w odniesieniu do nowego typu T_j jest możliwa dopiero po zmodyfikowaniu systemu.



Rys. 1. Przykład systemu z niezależnymi plikami bazy danych (wariant 1):
a) struktura plików bazy danych,
b) struktura systemu

W przytoczonym przykładzie ewidencjonowanie danych ilościowych może być dokonywane dla dowolnego wyrobu. Prowadzenie ewidencji danych specyficznych wymaga modyfikacji systemu.



Rys. 2. Przykład systemu z klasyfikacją obiektów (wariant 2):
a) struktura plików bazy danych,
b) struktura systemu

System z klasyfikacją obiektów

Realizacja tego wariantu wymaga wstępnej klasyfikacji wszystkich (także hipotecznych) obiektów ze zbioru W. Klasyfikacja ta musi doprowadzić do powstania hierarchii plików w sensie [3]. W drzewie hierarchii każdemu węzłowi odpowiada pewien typ, charakteryzowany przez podzbiór atrybutów. Najwyższemu węzłowi odpowiada podzbiór atrybutów, które charakteryzują każdy obiekt ze zbioru W. Reprezentantem takiej klasyfikacji jest hierarchia plików, w której plik najwyższy opisuje wszystkie obiekty, a pliki niższego poziomu odpowiednie podzbiory (rys. 2a). Procedury realizowane w systemie mogą być ogólne, przetwarzające pliki wyższego poziomu, oraz specyficzne – przetwarzające pliki tworzące ścieżki od korzenia do węzłów najniższych.

Wariant ten jest bardziej złożony i mniej sprawny. Jego wrażliwość na zmiany w asortymencie typów może być mniejsza niż wrażliwość wariantu pierwszego. Przykładowo, jeżeli pojawi się nowy typ wyrobu, trzeba będzie zdefiniować nowy plik podrzędny *WYRÓB 4*, przy nie zmienionym pliku *WYRÓB* i procedurach go przetwarzających. W zasadzie można tak skonstruować system, by mógł być częściowo użytkowany bez modyfikacji w odniesieniu do nowego typu T_j.

System z plikiem definicyjnym

Realizacja tego wariantu wymaga utworzenie dwóch plików: definicyjnego i danych. Pierwszy z nich zwiera definicję atrybutów każdego typu, a drugi – dane, których interpretacja zależy od informacji zawartych w pliku difinicyjnym. Procedury przetwarzania pliku danych są więc sparametryzowane i korzystają z pliku definicyjnego. Na rysunku 3a przedstawiono przykład plików dla założeń przyjętych na wstępie. W rzeczywistym systemie opis każdego atrybutu w pliku definicyjnym zawiera także obraz atrybutu (*PICTURE*) oraz kod sposobu przetwarzania atrybutu. Wariant ten wydaje się nie bardziej złożony niż poprzedni. Jego sprawność jest także porównywalna ze sprawnością poprzedniego wariantu (zależy od konkretnej realizacji idei z rys. 3). Natomiast idea uniwersalizacji sposobów pamiętania, przetwarzania oraz prezentacji danych w tym wariancie gwarantuje niewrażliwość systemu na zmianę asortymentu wyrobów, oczywiście w granicach tej uniwersalności. Pojawienie się nowego typu wyrobu wymaga wprowadzenia jego definicji do pliku definicyjnego. Obsługa pliku definicyjnego jest zwykłą funkcją użytkową. Dostosowanie systemu do nowego asortymentu odbywa się bez udziału projektantów i nie stwarza problemów organizacyjnych.

a)

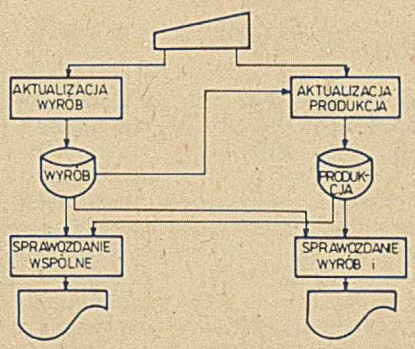
WYRÓB (PLIK DEFINICYJNY)

Symbol wyrobu	Ilość atrybutów	Atrybut 1	Atrybut 2	Atrybut 3
Wyrób 1	3	Wsad	Elektrody	Energia
Wyrób 2	1	Czas pracy wyparek	—	—
Wyrób 3	1	Czas pracy suszarek	—	—
⋮	⋮	⋮	⋮	⋮

PRODUKCJA (PLIK DANYCH)

Symbol wyrobu	Data	Ilość	Atrybut1	Atrybut2	Atrybut 3
Wyrób 1	88.01.02	150	200	15	20
Wyrób 2	88.01.02	20	6	—	—
Wyrób 3	88.01.02	35	7	—	—
⋮	⋮	⋮	⋮	⋮	⋮

b)



Rys. 3. Przykład systemu z plikiem definicyjnym (wariant 3):
a) struktura plików bazy danych,
b) struktura systemu

W zrealizowanym przez autora systemie zastosowano wariant trzeci, jako praktycznie niewrażliwy na zmianę asortymentu wyrobów i wykonalny dzięki uzgodnieniu z użytkownikami pewnych założeń ograniczających. Najistotniejsze były założenia dotyczące sposobu przetwarzania danych o poszczególnych typach wyrobów. Przyjęto więc, że wszystkie typy wyrobów podlegają jednakowej procedurze przetwarzania, polegającej na agregowaniu wartości każdego atrybutu z osobna. Pozostałe założenia były istotne raczej z punktu widzenia nakładów na realizację i z punktu widzenia sprawności, nie miały natomiast zasadniczego charakteru. Aby uprościć projektowanie i oprogramowanie form prezentacji, przyjęto ograniczenie na maksymalną liczbę atrybutów wyrobów (w rzeczywistym systemie 20). Założenie to stworzyło także podstawę do wyboru struktury plików: definicyjnego i danych. Zrealizowano pliki z rekordami „pseudozmiennej" długości jako wariant w pewnych zastosowaniach bardziej oszczędny pod względem pamięci i czasu dostępu.

W przedstawionym przykładzie uniwersalizacja fragmentu (wybranej funkcji) systemu jest konieczna przynajmniej z punktu widzenia użytkownika, który płaci za każdą adaptację systemu nieuniwersalnego. Przytoczone warianty w różnym stopniu gwarantują wrażliwość systemu na zmiany składu zbioru jednorodnego. Wariant drugi ma raczej teoretyczne znaczenie, ponieważ może być stosowany tylko w sytuacjach, w których dysponuje się informacją umożliwiającą dokonanie wstępnej klasyfikacji. Posiadanie tej informacji nie chroni jednak przed modyfikacją systemu.

LITERATURA

[1] Date C. J.: Wprowadzenie do baz danych. WNT, Warszawa 1981

[2] Rekuć W.: Opis i instrukcja eksploatacji systemu wspomagającego planowanie produkcji w ZDHM Legnica. Raport Instytutu Organizacji i Zarządzania Politechniki Wrocławskiej, seria SPR nr 3, Wrocław 1989

[3] Smith J. M., Smith D. C. P.: Database Abstractions Aggregation and Generalization. ACM TODS, Vol 2, No. 2, 1977.

INTERSOFT Sp. z o.o.

INTERSOFT

Nasze Biuro Handlu Zagranicznego poleca swoje usługi w zakresie eksportu-importu każdego towaru.
Prowadzi sprzedaż profesjonalnego sprzętu komputerowego i peryferii.

1. JĘZYKI PROGRAMOWANIA, BIBLIOTEKI, GRAFIKA

Basic – programowanie w języku Basic cz. III	52 000 zł
Turbo Basic	183 000 zł
Proffesional Fortran + dodatki	144 000 zł
Fortran 77	132 600 zł
Fortran 80	65 300 zł
Język C – zastosowanie dla zaawansowanych	214 200 zł
Kompilator języka C	106 000 zł
System Turbo C V.1.5	79 000 zł
Turbo C biblioteka	150 000 zł
Turbo C podręcznik użytkownika	109 000 zł
Turbo Pascal	123 500 zł
Turbo Pascal V.4.0	277 000 zł
Turbo Pascal V.5.0	161 000 zł
Turbo Pascal V.5.5 dodatek do T.P.V.5.0	56 000 zł
Turbo Power Tools Plus	124 000 zł
Turbo Database Toolbox	33 000 zł
Modula 2 86	125 500 zł
Turbo Pascal 5.0 opis języka	189 000 zł
Turbo Pascal 4.0 (aut. Jan Bielecki)	153 000 zł

2. BAZY DANYCH, PAKIETY ZINTEGROWANE

Clipper do dBase III	30 000 zł
Clipper do dBase III+	131 000 zł
Clipper 86	158 000 zł
Clipper 87	212 700 zł
Clipper 87 do dBase III+	214 000 zł
dBase II i III – podręcznik dla zaawansowanych	205 000 zł
dBase III + instalacja pakietu relacyjnej bazy danych	22 000 zł
dBase III + zastosowanie	205 000 zł
dBase III + opis pakietu sieciowego	55 000 zł
dBase III + programowanie	126 000 zł
dBase III + poznawanie	111 000 zł
dBase III – poradnik encyklopedyczny	237 000 zł
dBase IV	249 000 zł
Framework II P	162 000 zł
Informix	261 000 zł
Lotus 1-2-3 + dodatek (komplet)	113 000 zł
Multiplan	66 000 zł
Pakiet C-ISAM	34 700 zł
CX – Fortran kompilator języka do systemu D0S2	114 300 zł
Plib 86/Plink 86 opis języka	47 000 zł
Statgraphics	91 000 zł
Programowanie systemów baz danych języka Clipper	107 000 zł
Kompilator CBasic tom. II	92 000 zł

3. SYSTEMY OPERACYJNE, PROGRAMY UŻYTKOWE

System operacyjny DOS cz. 1 i 2	155 000 zł
System operacyjny DOS 3.10	47 000 zł
System operacyjny DOS 3.20	181 000 zł
System operacyjny DOS 3.30	180 000 zł
System operacyjny DOS 4.0	202 000 zł
Eureka	64 000 zł
Norton Commander	63 000 zł
Polywindows	50 000 zł
Sidekick	44 000 zł

4. EDYTORY

Chl-2 Writer	31 000 zł
Personalny Edytor PE	68 300 zł
WordStar 2000	142 000 zł

5. CAD-SIECI

AutoCad 2.17	137 000 zł
AutoCad	30 600 zł
Lanlink 4.0 sieć lokalna	80 000 zł
NOWELL, instalacja sieci Arcnet	9 000 zł
NOWELL, instalacja sieci	65 000 zł
NOWELL, podręcznik użytkowania sieci	116 000 zł
Or-Cad	134 000 zł

6. ROZMAITOŚCI

Przewodnik programisty	205 000 zł
Printer STAR NL-10 instrukcja	20 000 zł
Wprowadzenie do użytkowania komputerów osobistych klasy IBM PC	43 000 zł
Drukarka NX-16 instrukcja	106 000 zł
Drukarz	14 000 zł
Karta – system rozliczania kart drogowych	10 000 zł
Appegen	18 000 zł

Zamówienia prosimy kierować pod adresem:
INTERSOFT Sp. z o.o.
00–496 Warszawa, ul. Krucza 16/22,
telefony: 28-44-81 w. 284, 352; 28-29-53,
teleks: 812414

0/7/90

WITOLD R. RUDOLF
Otmuchów

System autorski JU-LEK

Od ponad 25 lat trwają prace nad zastosowaniem komputerów do wspomagania nauczania (ang. *Computer Assited Instruction – CAI*). Główne zalety CAI to: rozszerzenie możliwości prezentacji (symulacja i obserwacja przebiegów zjawisk dla różnych parametrów) oraz indywidualizacja nauczania (dostosowanie zakresu i tempa przekazywanego materiału do indywidualnych predyspozycji i potrzeb ucznia).

Pierwszym systemem nauczania wspomaganego komputerem był system *PLATO* (ang. *Programmed Logic for Automatic Teaching Operation*), opracowany w Computer-Based Education Research Laboratory wspólnie z firmą komputerową Control Data Corporation. *PLATO* jest największym obecnie tego rodzaju systemem. Początki prac nad nim sięgają końca lat pięćdziesiątych; w 1981 roku działało trzynaście takich systemów (osiem w USA, po jednym w Kanadzie, Australii i Południowej Afryce oraz dwa w Europie), obejmujących kilkanaście tysięcy lekcji.

SYSTEMY AUTORSKIE

W ostatnich latach na świecie szczególnie popularną dziedziną CAI stały się programy edukacyjne tworzone w postaci zestawów lekcji typu podającego (ang. *Presentation Type CAI – PT-CAI*) – ten model nauczania przejęto z systemu *PLATO*. Programy-lekcje (ang. *coursewa-re*) przekazują uczniowi materiał w postaci „ruchomej, inteligentnej elektronicznej książki z obrazkami", która potrafi zadawać pytania, oceniać odpowiedzi i odpowiednio na nie reagować. Lekcje takie powinna cechować dynamika – animacja, symulacja, dialog z uczniem itp. Do opracowywania zestawów takich lekcji powstały wyspecjalizowane narzędzia – **systemy autorskie.** Stanowią one pewien rodzaj bardzo prostych języków służących do tworzenia i wykonywania zestawów lekcji.

Podstawową zasadą systemów autorskich jest udostępnienie narzędzi pozwalających sprawnie i wygodnie konstruować oprogramowanie edukacyjne (lekcje) wszystkim zainteresowanym oraz rozdzielenie funkcji: tworzenia opisu lekcji i ich wykonywania.

Autor programu edukacyjnego nie musi umieć programować, przez co znacznie poszerza się krąg potencjalnych autorów lekcji. Mogą się oni przy tym skoncentrować wyłącznie na tworzeniu części dydaktycznej lekcji, nie zajmując się opracowywaniem programu prezentującego materiał dydaktyczny.

Autor lekcji tworzy jedynie dane, które można o wiele łatwiej pielęgnować niż programy. Łatwiej jest więc dostosować lekcję opracowaną przez jednego autora do konkretnych potrzeb użytkownika. Poza tym lekcja składająca się tylko z danych wymaga o wiele mniej pamięci niż odpowiadający jej zawartością dydaktyczną program napisany za pomocą jednego z języków programowania (dzięki temu stało się możliwe zastosowanie nawet komputerów 8-bitowych).

Autor konstruuje lekcję (właśnie konstruuje, a nie pisze program) korzystając z wcześniej opracowanego scenariusza lekcji. Co istotne, autor lekcji (konstruujący lekcję) i autor scenariusza nie musi być tą samą osobą – prostota konstruowania lekcji zapewnia łatwe zrealizowanie szczegółowego scenariusza przez osobę obsługującą system i łatwe wprowadzanie poprawek.

System autorski umożliwia konstruowanie lekcji z prostych elementów umieszczanych kolejno na ekranie (np. tekstów, rysunków lub animacji tych elementów) oraz zadawanie pytań i ocenianie odpowiedzi podanej przez ucznia (system powinien odpowiednio reagować wyświetlając komunikat, powtarzając lub opuszczając pewien fragment materiału – sposób reakcji określa autor). System ułatwia tworzenie i testowanie przygotowywanej lekcji, jest przyjazny (ang. *user friendly*). Uczeń pracując z lekcją opracowaną za pomocą systemu autorskiego ma możliwość częściowego (w zakresie ustalonym przez autora w trakcie tworzenia lekcji) wpływania na przebieg lekcji i dopasowywania tempa podawania informacji do swoich możliwości. W szczególności może:

- wybrać temat w ramach lekcji ze spisu treści,
- zadecydować, kiedy system ma przedstawić następny fragment materiału,
- kontynuować prezentację, pominąć bądź powtórzyć fragmenty materiału,
- skontrolować wiedzę – system zadaje pytania i ocenia odpowiedzi ucznia podając w razie potrzeby dodatkowe informacje lub powtarzając (pomijając) pewną partię materiału.

Klasycznym przykładem systemu autorskiego jest system Autool, opracowany pod kierunkiem prof. dr Hermanna Maurera na Uniwersytecie Technicznym w Graz (Austria). System ten opracowano dla 8-bitowego mikrokomputera MUPID o parametrach zbliżonych do Amstrada 6128. Istnieje również wersja systemu Autool dla komputerów klasy IBM PC. Lekcje opracowane w systemie Autool mogą być rozprowadzane na dyskietkach lub w ogólnokrajowej sieci wideotekstu. W 1986 roku austriacka baza lekcji dostępnych w sieci wideotekstu liczyła ponad 600 jednostek lekcyjnych, korzystało z niej ponad 12 tys. użytkowników. Za pomocą systemu Autool jest obecnie realizowany międzynarodowy projekt *COSTOC* (ang. *Computer Supported Teaching of Computer Science*) mający na celu opracowanie zestawów lekcji do nauczania informatyki na poziomie uniwersyteckim.

SYSTEM AUTORSKI DLA KOMPUTERA ELWRO 800 JUNIOR

Przeniesienie idei systemów autorskich do polskich szkół wydawało się bardzo nęcące, ze względu na fakt, że za pomocą takich systemów lekcje mogą być przygotowywane przez nauczycieli nie znających żadnego języka programowania. Realizacją tego zamierzenia jest **system do tworzenia i wykonywania zestawów lekcji JU-LEK** (skrót od JUnior i LEKcja), opracowany w ramach pracy dyplomowej autora tego tekstu, napisanej w Instytucie Informatyki Uniwersytetu Wrocławskiego. Praca otrzymała III nagrodę w Ogólnopolskim Konkursie Prac Magisterskich z Informatyki organizowanym przez PTI. System JU-LEK był prezentowany podczas V Krajowej Konferencji „INFORMATYKA W SZKOLE" w Wałbrzychu, we wrześniu 1989 r.

System JU-LEK, wzorowany na systemie Autool, został opracowany dla mikrokomputerów ELWRO 800 Junior pracujących pod kontrolą systemu operacyjnego CP/J w wersji 2.23. System jest prosty w obsłudze i umożliwia sprawne konstruowanie lekcji. Ponieważ **JUNIORY** mogą pracować w sieci lokalnej JUNET, JU-LEK umożliwia korzystanie z gotowych lekcji za pośrednictwem tejże sieci.

Skonstruowanie lekcji polega na umieszczeniu na ekranie obiektów, które mają być zaprezentowane uczniowi. System jest przyjazny dla użytkownika – podczas tworzenia lekcji prowadzi autora przez swoje kolejne poziomy, zadaje dodatkowe pytania, ułatwia testowanie i modyfikowanie przygotowanej lekcji.

Podstawowym elementem lekcji jest, podobnie jak w systemie Autool ramka (ang. *frame*). Lekcja może składać się co najwyżej z 99 ramek. Ramkę można interpretować jako logiczną jednostkę nauczania składającą się z informacji tworzących spójną całość. W trakcie wykonywania lekcji ramka jest zawsze całkowicie przerabiana. Zdefiniowane połączenia między ramkami tworzą *graf przebiegu lekcji* (rys. 1). Ramki są pamiętane na dyskietce w postaci plików – każdej ramce odpowiada plik o nazwie zgodnej z nazwą lekcji i rozszerzeniu będącym numerem ramki. Dodatkowy plik o nazwie **NAZWA _ LEKCJI. 00** zawiera informacje o typach ramek i połączeniach między nimi, plik **NAZWA _ LEKCJI. ODP** zawiera dane o obiektach ODPOWIEDŹ. Ramki są prezentowane uczniowi w kolejności określonej przez autora lekcji. Elementy

składowe ramki pojawiające się na ekranie, czyli np. fragmenty tekstu, elementy graficzne, przerwy czasowe, nazwano **obiektami**. Ograniczenia pamięci powodują, że jedna ramka może liczyć co najwyżej 150 obiektów. Można ominąć to ograniczenie przez nałożenie na siebie dwóch lub więcej ramek. Sposób ten pozwala także wykorzystać jeden rysunek w kilku kolejnych ramkach. Obiekty w ramce są pamiętane w postaci listy – istotna jest kolejność, w jakiej zostały umieszczone w ramce.

Tworzenie lekcji polega na redagowaniu zawartości poszczególnych ramek i ustaleniu połączeń między nimi. Tworzenie ramki polega na kolejnym definiowaniu obiektów na ekranie przez wskazanie kursorem pozycji obiektu i ewentualnie podanie jego parametrów w odpowiedzi na dodatkowe pytania systemu. Można także modyfikować zawartość wcześniej utworzonej ramki – usuwając, przemieszczając lub powielając obiekty. Każdy utworzony obiekt jest natychmiast wyświetlany na ekranie. W systemie są dostępne następujące obiekty:

TEKST – zawiera do 62 znaki; w trybie 64-znakowym (normalne) lub 32-znakowym (powiększone), można wyświetlać znaki na jasnym tle;
KASUJ – czyści (kasuje) fragment ekranu;
PAUZA – wstrzymuje na podany czas wyświetlanie obiektów;
POWTÓRZ – wielokrotnie powtarza sekwencję obiektów;
PASCAL – uruchamia zewnętrzny program w Turbo Pascalu 3.0;
WSKAŹNIK – pozwala utworzyć w ramce spis treści;
ODPOWIEDŹ – pobiera i analizuje odpowiedzi ucznia oraz określa reakcję systemu.

Obiekty graficzne

LINIA – łamana licząca do 10 wierzchołków,
OKRĄG – okrąg lub koło (zależnie od typu wypełnienia),
PROSTOKĄT – prostokąt,
WEKTOR – wektor.

Obiekty – atrybuty obiektów graficznych

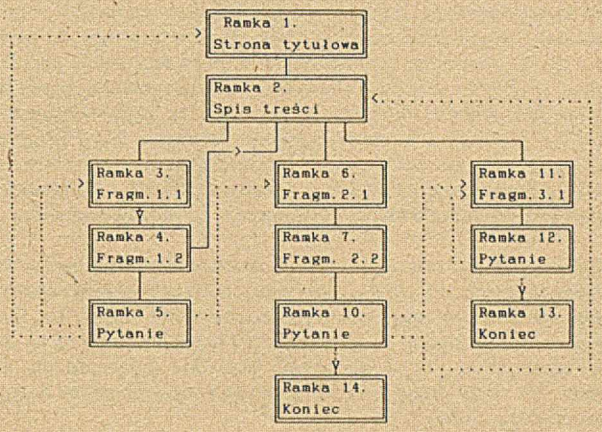
RODZAJ LINII – określa wzorzec rysowanych linii (np. ciągła, przerywana, kropkowana itp.),
KOLOR – ustala kolor, którym są rysowane figury, jako: czarny (tła), biały (atramentu) lub odwracający (zapala zgaszony, a gasi zapalony punkt),
WYPEŁNIENIE – określa typ wypełniania dla prostokątów i okręgów (pełny, pusty, wypełniony wzorem),
WNĘTRZE – określa wzór wypełniania prostokątów i okręgów (linie pionowe, ukośne, kratka).

Numerami oznaczono następujące obiekty:

```
1 - LINIA
2 - PROSTOKĄT z atrybutami:
        RODZAJ LINII = ......
        WYPEŁNIENIE = pusty
        KOLOR = atramentu
3 - WEKTOR z atrybutem RODZAJ LINII = - - -
4 - PROSTOKĄT z atrybutami:
        WYPEŁNIENIE = pełny
        KOLOR = odwracający
    narysowany na obiekcie 1
5 - PROSTOKĄT jak w 2 narysowany wokół obiektu TEKST
6 - OKRĄG z atrybutami:
        WYPEŁNIENIE = wzorem
        WNĘTRZE = linie pionowe
        KOLOR = atramentu
7 - OKRĄG z atrybutami:
        WYPEŁNIENIE = pełny
        KOLOR = atramentu
8 - OKRĄG z atrybutami:
        WYPEŁNIENIE = wzorem
        WNĘTRZE = linie skośne w lewo
        KOLOR = atramentu
9 - kursor graficzny
10- TEKST (duże znaki na jasnym tle)
11- TEKST (normalne znaki)
```

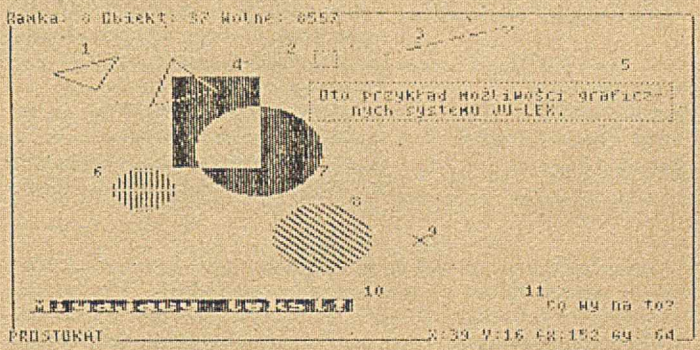
Obiekty rozmieszcza się na ekranie wskazując ich pozycję kursorem. Ramki można nakładać na siebie lub kopiować zawartość jednej ramki do innej, co pozwala np. wykorzystać jeden rysunek w kilku ramkach i prowadzi do znacznej oszczędności czasu podczas konstruowania lekcji.

Mimo niewielkiego zestawu obiektów można z nich budować złożone ramki, a także uzyskać prostą animację, kolejno wyświetlając obiekt i gasząc go. Szczególnie istotna jest możliwość kontrolowania stopnia przyswojenia wiedzy przez ucznia i określenie reakcji systemu w zależności od odpowiedzi ucznia na zadane pytanie. Zadanie to realizuje obiekt ODPOWIEDŹ.



Rys. 1. Przykład możliwych połączeń między ramkami

Z każdą ramką są związane co najmniej dwa wskaźniki do innych ramek: poprzedniej i następnej. Tworzą one sieć połączeń między ramkami (rys. 1), po której może wędrować uczeń. Autor może pozwolić uczniowi (po zaprezentowaniu przez system materiału z danej ramki) na przejście do ramki następnej lub poprzedniej, przy czym wskaźnik do ramki poprzedniej nie koniecznie musi wskazywać na ramkę bezpośrednio wcześniej wykonywaną (por. połączenie między ramkami 3 i 4 na rys. 1).

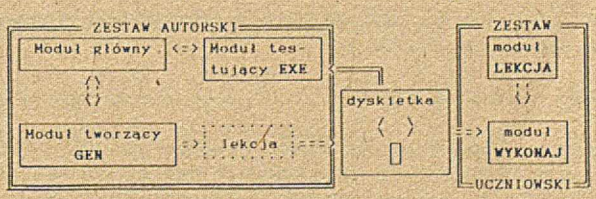


Rys. 2. Przykłady obiektów

Za pomocą obiektu WSKAŹNIK można w ramce zbudować rodzaj „spisu treści", który pozwoli uczniowi wybrać interesujący go fragment materiału. W ten sposób można podzielić materiał np. na rozdziały i podrozdziały powiązane tematycznie. Pojedynczy obiekt WSKAŹNIK (w ramce może wystąpić tylko jeden taki obiekt) pozwala zdefiniować do 8 dodatkowych wskaźników do ramek. Na rysunku 1 obiekt ten znajduje się w ramce 2.

BUDOWA SYSTEMU JU-LEK

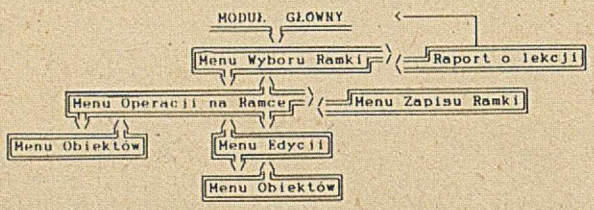
Schemat budowy systemu pokazano na rysunku 3. Ze względu na ograniczoną wielkość dostępnej pamięci komputera **JUNIOR** przyjęto modułową strukturę systemu.



Rys. 3. Schemat budowy systemu JU-LEK

Moduły: tworzący **GEN** i testujący **EXE** są uruchamiane przez moduł główny, który jest uruchamiany z poziomu systemu operacyjnego CP/J. Również z poziomu systemu CP/J jest uruchamiany moduł **LEKCJA** zestawu uczniowskiego, który automatycznie uruchamia moduł **WYKONAJ** tego zestawu. Zestaw autorski i zestaw uczniowski stanowią niezależne, samodzielne części systemu. Zestaw autorski jest przeznaczony dla komputerów wyposażonych w prywatną stację dysków, ze względu na częste odwołania do dysku podczas opracowywania lekcji; możliwe jest również konstruowanie lekcji za pośrednictwem sieci JUNET – wówczas można przyspieszyć tworzenie lekcji przez pracę równoległą, tzn. np. kilka osób jednocześnie buduje oddzielne ramki według scenariusza tej samej lekcji. Pojedyncze ramki można następnie bardzo szybko zmontować na komputerze zarządzającym siecią.

Moduł **GEN** jest obsługiwany za pomocą wielopoziomowego menu. Podczas tworzenia lub poprawiania ramki bieżącą postać ekranu można zapisać w pliku o standardowym rozszerzeniu COD, co ułatwia tworzenie dokumentacji lekcji podczas jej konstruowania. Z tej możliwości skorzystano, by wydrukować wszystkie dołączone do tekstu rysunki. Na rys. 4 przedstawiono schemat połączeń między kolejnymi menu modułu GEN.



Rys. 4. Schemat połączeń między kolejnymi menu modułu GEN

Kolejne menu umożliwiają następujące operacje:

Menu Wyboru Ramki	– wybór bieżącej ramki;
Menu Operacji na Ramce	– wybór operacji wykonywanej na całej ramce (redagowanie, dodanie obiektu, skopiowanie z zapisanie ramki itp.);
Menu Obiektów	– dołączanie nowych obiektów;
Menu Edycji	– dokonywanie zmian obiektów;
Menu Zapisu Ramki	– zapisanie ramki na dyskietkę;
Raport o lekcji	– przechowywanie informacji o operacjach wykonanych na ramkach podczas tworzenia lekcji.

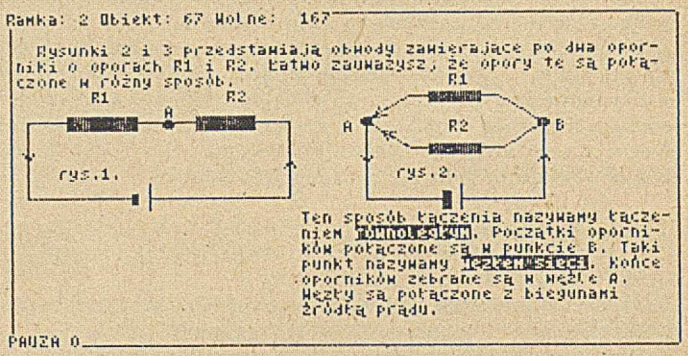
Autor lekcji podejmuje decyzję wybierając jedną z operacji dostępnych w danym menu. W razie konieczności JU-LEK żąda podania dodatkowych informacji dla systemu. W przypadku operacji krytycznych (usunięcie obiektu, skasowanie ramki, zakończenie definiowania nowego obiektu itp.) system prosi autora o potwierdzenie decyzji.

Moduł testujący **EXE** jest przeznaczony do testowania lekcji utworzonej za pomocą modułu **GEN**. Moduł **EXE** pozwala prześledzić sposób wyświetlania kolejnych obiektów w ramce. W polu informacyjnym są wyświetlane dodatkowe informacje dla autora (numer ramki, numer wyświetlanego obiektu i jego nazwa). Podczas wyświetlania obiektów można przerwać test naciskając klawisz **ESC**. Poza tym ramki są wykonywane dokładnie tak, jak będą przedstawiane uczniowi podczas wykonywania lekcji przez zestaw uczniowski.

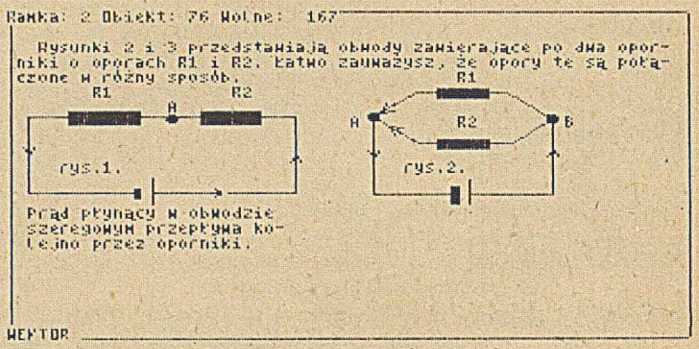
Zestaw uczniowski jest przeznaczony wyłącznie do wykonywania lekcji przygotowanych za pomocą systemu JU-LEK. Może pracować w sieci lokalnej JUNET (komputery uczniowskie korzystają wtedy ze wspólnej stacji dysków). Podobnie jak moduł **EXE**, pobiera on i interpretuje dane o lekcji przygotowane za pomocą zestawu autorskiego.
Zestaw uczniowski jest samodzielnym dwumodułowym programem i może być używany oddzielnie.

LEKCJA PRZYKŁADOWA

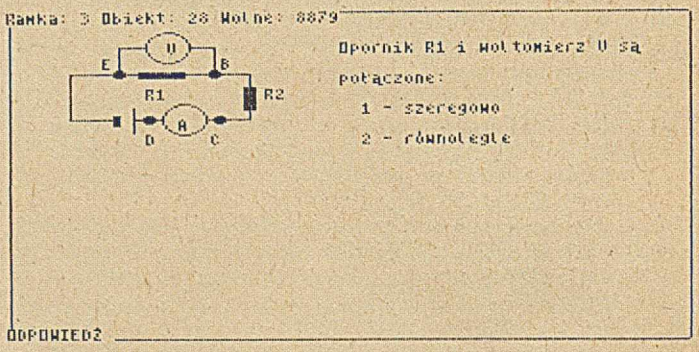
Poniżej przedstawiono kopie ekranu wykonane podczas tworzenia dwóch niewielkich lekcji, przedstawiających rodzaje łączenia oporników (FIZYKA; rys. 5–7) i sortowanie bąbelkowe (BUBBLE; rys. 8–13). W tych przykładach postać ekranu ulega częściowym zmianom praktycznie w każdej ramce. Występuje tu również animacja – jest ilustrowany przepływ prądu w obwodzie (rys. 6) i przesuwanie na właściwe pozycje kolejno porównywanych elementów ciągu. Rysunki nie oddają oczywiście tych efektów. Możliwość ocenienia wiedzy ucznia przedstawiono na rys. 7 – po zadaniu testowego pytania (można zdefiniować do 8 wersji odpowiedzi) system będzie oczekiwać na wybranie poprawnej odpowiedzi.



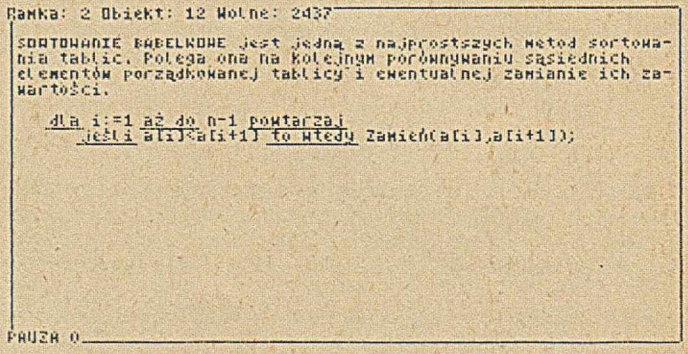
Rys. 5. Przykładowa postać ramki



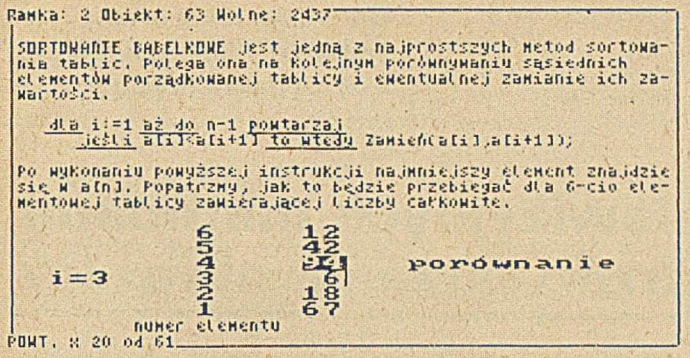
Rys. 6. Częściowe zmiany w ramce



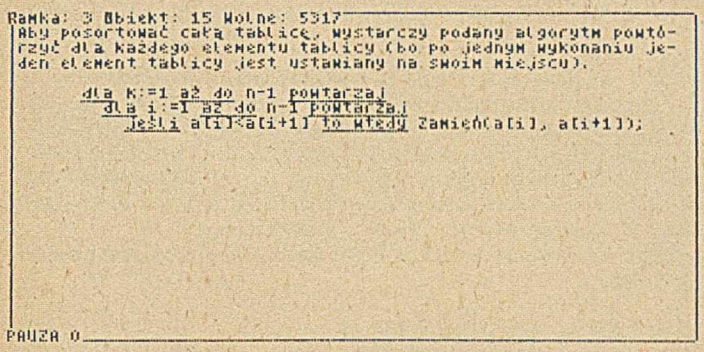
Rys. 7. Pytanie



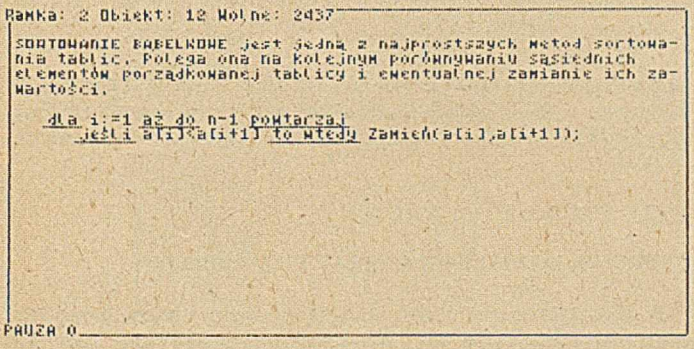
Rys. 8. Początek prezentacji



Rys. 9. Porównanie elementów

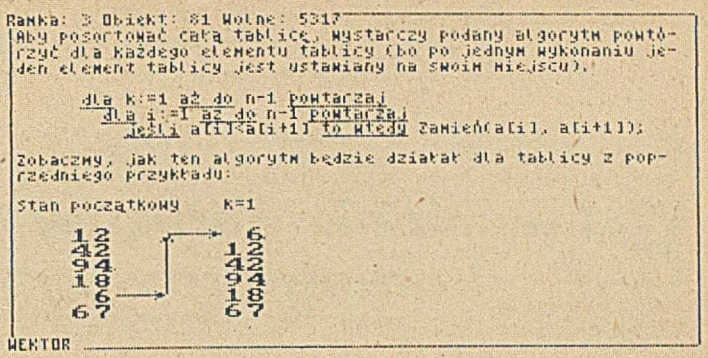


Rys. 10. Zamiana elementów

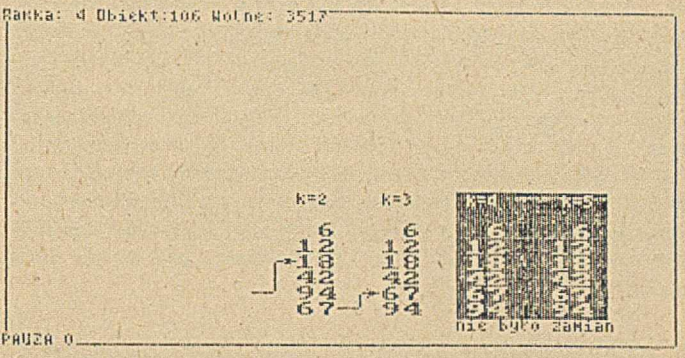


Rys. 11. Ciąg dalszy prezentacji

Przykład ten oczywiście nie demonstruje pełnych możliwości systemu. JU-LEK pozwala równie łatwo przedstawić np. dynamicznie zmieniającą się mapę granic Rzeczypospolitej lub rozwój rośliny – począwszy od nasiona.



Rys. 12. Koniec ramki



Rys. 13. Ramka nakładana na poprzednią

LITERATURA

[1] Bielecki J.: Turbo Pascal 3.0. WNT, Warszawa 1988

[2] Cellary W.: Podręcznik użytkownika mikrokomputera ELWRO 800 Junior. System operacyjny CP/J. WNT, Warszawa 1988

[3] Garratt J., Mauer H.: Autool Version 2. Manual for COSTOC Authors. Report 244, Berichte Institut für Informationsverarbeitung, Technische Universität Graz und Osterreichische Computer Gesellschaft, November 1987

[4] Huber F., Maurer H.: Extended Ideas on Editors for PT-CAI. Report 240, Berichte Institut für Informationsverarbeitung, Technische Universität Graz und Osterreichische Computer Gesellschaft, August 1987

[5] Kott R.: Programowanie w języku Pascal. PWN, Warszawa 1989

[6] Makedon F., Maurer H., Ottomann T.: Presentation type CAI in computer science education at university level. Journal of Microcomputer Applications, No.10, pp.283–295

[7] Papst A.: Project COSTOC – wspomagane komputerem nauczanie na poziomie szkoły wyższej. Techniki komputerowe, nr 3–4, 1988

[8] Papst A.: System autorski AUTOOL. Techniki komputerowe, nr 1, 1989

[9] Stubenrauch R., Maurer H.: A General Lesson Specification System. Report 241, Berichte Institut für Informationsverarbeitung, Technische Universität Graz und Osterreichische Computer Gesellschaft, September 1987.

Lubelskie Przedsiębiorstwo Informatyki Przemysłu Budowlanego ETOB w Lublinie

posiada do sprzedaży:

- **tuleje drukujące (bębny) do DW-304 – 120 znakowe**
- **drukarki znakowo-mozaikowe D-180-KSRE**

Urządzenia są nowe fabrycznie – cena do uzgodnienia.

Informacje pod numerem telefonu 235-71 w Lublinie.

O/8/90

NGUYEN KIM SACH
Instytut Radioelektroniki
Politechnika Warszawska

Wzmacnianie obrazów telewizyjnych metodami komputerowego przetwarzania obrazów

Obrazy telewizyjne przesyłane na duże odległości są zwykle rozmazane, a część informacji obrazowych jest ukryta w szumach. Szumy są efektem przypadkowych fluktuacji elektrycznych, powstających wskutek ziarnistej struktury elementów elektronicznych bądź ładunków elektrycznych. Istnieją różne rodzaje szumów, np. szum cieplny, szum śrutowy i szum impulsowy [5]. W celu poprawienia jakości obrazów telewizyjnych należy je oczyścić z szumów i wzmocnić. Można w tym celu zastosować techniki przetwarzania obrazów opracowane dla potrzeb komputerowego przetwarzania informacji obrazowej. Stosuje się filtry liniowe i nieliniowe. Duże znaczenie mają w tym przypadku zwłaszcza filtry nieliniowe, np. filtr wielozakresowy (ang. *rank filter*) i filtr środkowy (ang. *median filter*) [6, 7].

Cyfrowe wzmacnianie obrazu dwuwymiarowego jest procesem przybliżonego obliczania obrazu oryginalnego. Wzmacnianie krawędzi w obrazie dwuwymiarowym ma natomiast na celu nadanie obrazowi postaci wygodnej do bezpośredniej obserwacji lub analizy maszynowej. Wzmacnianie krawędzi obrazu nie jest tożsame ze wzmacnianiem obrazu. W celu wzmacniania krawędzi obrazu często stosuje się filtry górnoprzepustowe. Istnieją różne techniki wzmacniania krawędzi obrazu, np. rozciąganie kontrastu, uwyraźnienie krawędzi lub manipulowanie skalą jasności w obrazie [4, 5, 6].

CYFROWE WZMACNIANIE DWUWYMIAROWEGO OBRAZU ZASZUMIONEGO

Przy przetwarzaniu obrazów dwuwymiarowych najczęściej stosuje się transformację Fouriera. Musi być ona realizowana w taki sposób, aby operacja filtracji dyskretnej była dokładnie reprezentowana przez operację splotu.

$$\hat{g}_1(x, y) = g_2(x, y) * h(x, y) \tag{1}$$

gdzie: $\hat{g}_1$ – funkcja oszacowania obrazu ciągłego g_1, g_2 – funkcja obrazu ciągłego (obserwowanego), h – funkcja dyspersji punktu, $*$ – operator splotu, a forma dyskretnej odpowiedzi impulsowej filtru wzmacniania obrazu dokładnie modelowała odpowiedź impulsową filtru analogowego [5]. Filtr środkowy (specjalny przypadek filtru wielozakresowego) jest sposobem nieliniowego przetwarzania sygnałów, stosowanym w celu zredukowania szumu w obrazie bez znacznego zakłócenia i zniszczenia krawędzi obrazu [1, 8]. W dwuwymiarowym filtrze środkowym jest stosowane dwuwymiarowe okno o określonym kształcie (np. kwadratu lub prostokąta) przesuwające się po obrazie. Środkowy piksel (element obrazu) okna jest wymieniany na wartość średnią jaskrawości piksli w oknie. Filtr środkowy eliminuje te piksle, dla których wartość intensywności znacznie odbiega od wartości intensywności pozostałych piksli w otoczeniu. W ten sposób można usuwać wszystkie piksle o największych intensywnościach, reprezentujące szum lub zakłócenie.

W obszarach o dużej liczbie szczegółów stosuje się filtr środkowy z małym oknem, np. okno 3×3 piksle. Natomiast na obszarach o mało różniących się jaskrawościach, wymiar okna może być zwiększony, np. do rozmiaru 5×5 piksli. Do wzmacniania jednego obrazu można stosować kaskadową filtrację środkową o stałym lub zmiennym wymiarze okna. Filtr środkowy jest skuteczniejszy przy redukowaniu szumu powodowanego impulsami dyskretnymi niż przy wygładzaniu szumu i ma możliwość znacznego zachowania krawędzi obrazu [1, 3].

Dwuwymiarowy filtr środkowy z oknem B na obrazie A jest definiowany jako średnia wartość intensywności wszystkich piksli w oknie B, zastępująca wartość środkową okna, co można zapisać

$$y_{ij} = \underset{B}{\text{środ}}(x_{ij}) \triangleq \underset{B}{\text{środ}}[x_{i+p,j+q}; (p,q) \in B] \tag{2}$$

gdzie: $A = \{x_{ij}; i,j = 1(1)N\}$, p, q – elementy wiersza i kolumny okna B, x_{ij} – wartości intensywności piksla obrazu wejściowego, y_{ij} – wartość intensywności piksla obrazu wyjściowego. Krawędź obrazu jest linią dyskretną [1], taką, że piksle po jednej jej stronie mają wspólną wartość jaskrawości l, a piksle po jej drugiej stronie – wartość jaskrawości h, przy czym l i h leżą w określonym przedziale wartości, oraz $l \neq h$. Różnica $d = l - h$ jest nazywana *wysokością krawędzi obrazu*.
Jeżeli zbiór wartości piksli okna B jest symetryczny względem punktu $(0, 0)$, tzn.

$$(l, h) \in B, (-l, -h) \in B \text{ i } (0, 0) \in B \tag{3}$$

to filtr środkowy

$$y_{ij} = \text{środ}\,[x_{i+l,j+h}, (l, h) \in B], \quad (i, j) \in N^2 \tag{4}$$

będzie zachowywał krawędzie obrazu [1]. Jeżeli krawędź K jest traktowana jako różnica wartości jaskrawości między dwoma pikslami we wszystkich kierunkach, to warunki zachowania krawędzi można określić jako

$$(l, h) \in B, (l, h) \in K \text{ i } d = l - h \neq 0 \tag{5}$$

CYFROWE WZMACNIANIE KRAWĘDZI OBRAZU

W celu zachowania i wzmacniania krawędzi obrazu mogą być stosowane adaptacyjne filtry statystycznie porządkowane, czyli OS-filtry (ang. *order statistic filter – OS filter*), filtry porównująco-wybierające, czyli CS-filtry (ang. *comparision and selection filter – CS filter*) oraz filtry środkowe [2]. Filtry środkowe mogą zachowywać, ale nie wzmacniają krawędzi obrazu. OS-filtr jest modyfikacją filtru środkowego. Może on wzmocnić krawędzie obrazu i redukować szumy impulsowe oraz niektóre części addytywnego szumu białego. CS-filtr może natomiast redukować impulsowy szum i niektóre części nieimpulsowego addytywnego szumu białego (gaussowskiego) przy wzmacnianiu krawędzi obrazu. CS-filtr można opisać zależnością

$$y_k = \begin{cases} x_k^{(N+1-J)}, \text{ jeżeli } \mu_k \geq M_k \\ x_k^{(N+1+J)}, \text{ w pozostałych przypadkach} \end{cases} \tag{6}$$

gdzie: y_k – wyjście CS-filtru, $x_k^{(i)}$ – i-ta próbka najmniejsza w oknie, $x_{k\,N}, \ldots, x_{k+N}$ – wartości wejściowe, $2N$ – długość okna, J – parametr o liczbie powtórzeń k, $\mu_k \cdot M_k$ – wartość uśrednienia i próbka uśredniająca, $1 \leq J \leq N$.

Jeżeli $J = N$, to CS-filtr przyjmuje minimalną lub maksymalną wartość intensywności w oknie i $y_k = x_k$. Jeśli $J = 0$, to CS-filtr staje się filtrem środkowym i $y_k = x_k^{N+1}$. Ponieważ CS-filtr jest nieliniowy, więc nie można opisać go w ogólnej postaci. Może on redukować dobrze szum impulsowy i nieimpulsowy, jeżeli J jest małe, na przykład równe 1. Jeśli natomiast J jest duże, to CS-filtr dobrze wzmacnia krawędzie obrazu.

W celu wzmacniania krawędzi obrazu stosuje się różne operatory, np. operator Laplace'a, operator Roberta itp. [1, 2]. Autor zajmował się

następującymi z nich: modułem gradientowym, trzykrokowym operatorem Roberta (rys. 1). Poniżej podajemy ich definicje.

● **Moduł gradientowy** jest reprezentowany przez wartość bezwzględną różnicy jaskrawości piksla danego i piksli na rogach okna 3 × 3 (rys. 1a):

$$y_{ijG} = |x_{ij} - x_{i+1,\,j+1}| + |x_{ij} - x_{i-1,\,j-1}| + |x_{ij} - x_{i+1,\,j-1}| + |x_{ij} - x_{i-1,\,j+1}| \quad (7)$$

● **Trzykrokowy operator wyprzedzający** jest reprezentowany przez różnicę między sumą jaskrawości trzech kolejnych (po danym) piksli w wierszu i potrojonej wartości jaskrawości piksla danego (rys. 1b):

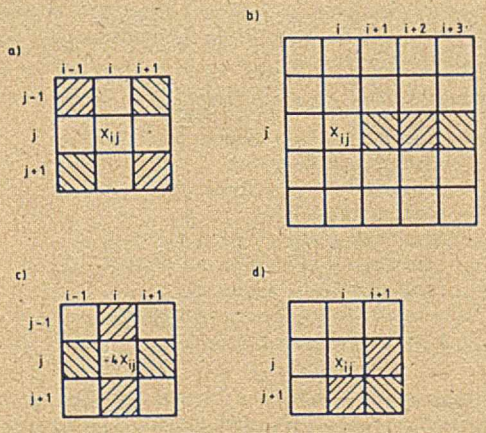
$$y_{ij3} = (x_{i+1,\,j} - x_{ij}) + (x_{i+2,\,j} - x_{ij}) + (x_{i+3,\,j} - x_{ij}) \quad (8)$$

● **Operator Laplace'a** jest reprezentowany przez różnicę między wartością sumy jaskrawości piksli sąsiednich z danym w wierszu i kolumnie a czterokrotną wartością jaskrawości piksla danego (rys. 1c):

$$y_{ijL} = (x_{i+1,\,j} + x_{i-1,\,j} + x_{i,\,j+1} + x_{i,\,j-1}) - 4x_{ij} \quad (9)$$

● **Gradientowy operator Roberta** jest reprezentowany przez sumę bezwzględnych wartości z różnic jaskrawości: piksla danego i piksla sąsiedniego w przekątnej czwartej ćwiartki oraz jaskrawości piksli sąsiednich z danym w czwartej ćwiartce (rys. 1d):

$$y_{ijR} = |x_{ij} - x_{i+1,\,j+1}| + |x_{i+1,\,j} - x_{i,\,j+1}| \quad (10)$$



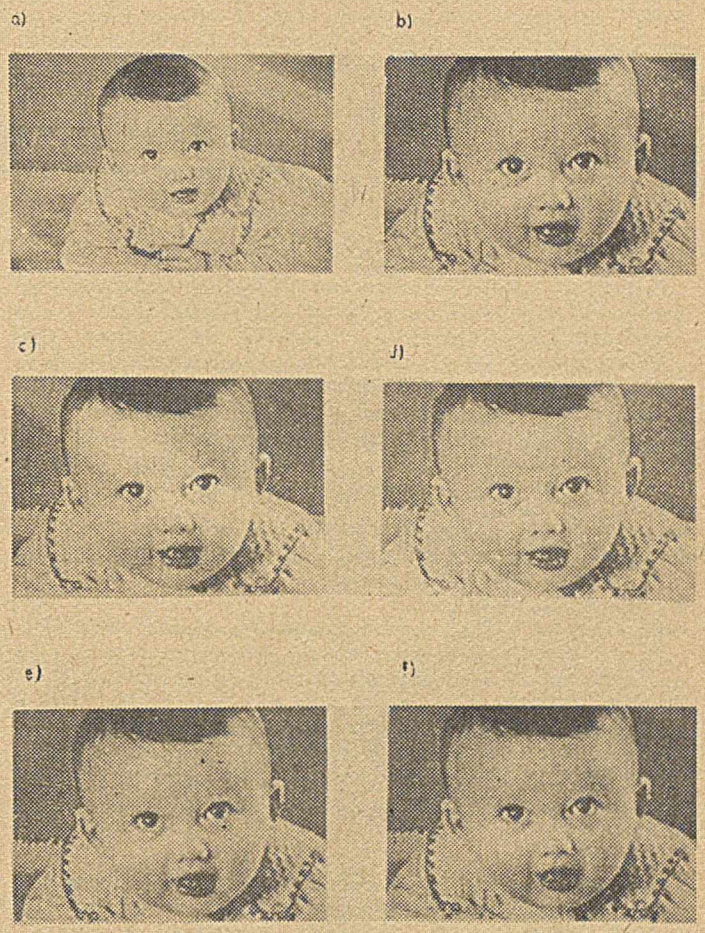
Rys. 1. Opis operatorów uwydatnienia krawędzi obrazu w oknie (a, b, c, d)

W eksperymencie, który autor przeprowadził, zastosowano połączenie operatora filtru środkowego i operatorów wzmacniania krawędzi obrazu, opisanych wzorami (7)–(10). Najpierw była realizowana operacja wzmacniania obrazu za pomocą filtru środkowego, a następnie operacja wzmacniania krawędzi obrazu za pomocą jednego z wymienionych operatorów. Stosowano różne typy szumów (np. szum o rozkładzie równomiernym, gaussowskim, Poissona, szum impulsowy) i różne obrazy typowe (np. obrazy o małych i dużych szczegółach oraz obrazy testowe, stosowane w telewizji). Wyniki są zadowalające w różnym stopniu. Na rysunku 2 przedstawiono wyniki programów realizujących operację filtru środkowego i modułu gradientowego (a), trzykrokowego operatora wyprzedzającego (b), operatora Laplace'a (c) i operatora Roberta (d) w wypadkach szumu o rozkładzie równomiernym i gaussoskim.

• • •

Do wzmacniania obrazów zaszumionych najlepiej jest stosować filtr środkowy [7]; do wzmacniania krawędzi obrazu można natomiast stosować moduł gradientowy, trzykrokowy operator wyprzedzający (są to adaptacyjne filtry środkowe), operator Laplace'a i operator Roberta. Kombinacja operatora wzmacniania obrazu i operatora wzmacniania krawędzi obrazu pozwala usunąć szum i jednocześnie poprawić krawędzie obrazu. Wyniki zastosowania tych operatorów są zadowalające.

Przedstawioną metodę można wykorzystać również do wzmacniania obrazów telewizyjnych w czasie rzeczywistym (np. przez podział obrazu na podobrazy i równoległe ich przetwarzanie za pomocą sprzętowych modułów przetwarzania podobrazu 64 × 64 piksle [5].



Rys. 2. Wyniki zastosowania operatorów wzmacniania obrazu i operator wzmacniania krawędzi obrazu;
a) obraz oryginalny,
b) obraz po dodaniu 10% szumu o rozkładzie gaussowskim,
c) obraz po zastosowaniu filtru środkowego i modułu gradientowego,
d) obraz po zastosowaniu filtru środkowego i trzykrokowego operatora wyprzedzającego,
e) obraz po zastosowaniu filtru środkowego i operatora Laplace'a,
f) obraz po zastosowaniu filtru środkowego i operatora Roberta

LITERATURA

[1] Huang T. S.: Two-dimensional digital signal processing. Vol. 2, Springer Verlag, Berlin 1981
[2] Lee Y. H., Fam A. T.: An edge gradient enhancement adaptive order statistic filter. IEEE Trans. on acoustics, speech, and signal processing. Vol. ASSP-35, No. 5, pp. 680 695, 1987
[3] Pratt W. K.: Digital image processing. New York 1978
[4] Sach N. K.: Szumy i odtwarzanie obrazów telewizyjnych metodami cyfrowymi. Przegląd techniki. Radio i Telewizja, nr 2, s. 12 22, 1989
[5] Sach N. K.: Poprawa jakości obrazów telewizyjnych metodami cyfrowymi. Prace Naukowe, Politechnika Warszawska, z. 91 (w druku)
[6] Sach N. K.: Zmniejszanie zakłóceń obrazów za pomocą filtru medianowego. Kwartalnik Elektroniki i Telekomunikacji, PAN, nr 3 (w druku)
[7] Sach N. K.: Odtwarzanie cyfrowe obrazów przy pomocy modyfikującego filtru medianowego. Krajowe Sympozjum Telekomunikacji KST '89. Bydgoszcz, 6–8 września 1989
[8] Stanke G., Osten W.: Rangordnungsoperatoren auf einem parallelen Bildprozessor. Bild und Ton. No. 3, pp. 84 88, 1989.

Przypominamy!
Do końca sierpnia można jeszcze zamówić prenumeratę INFORMATYKI na czwarty kwartał.

LUBOMIR JURCZAK
WOJCIECH WARSKI
Przedsiębiorstwo Zagraniczne COMPUTEX
Warszawa

System baz danych CX-DMOS (2)

Problemami o podstawowym znaczeniu w tworzeniu każdego oprogramowania użytkowego jest szybkość wykrywania błędów i czas wprowadzania wynikających stąd poprawek. Cechy systemu CX-DMOS, takie jak wielodostępność/wieloprogramowość, systemowa kontrola dostępu do bazy danych itp., powodują, że uruchomienie programów użytkowych „od zera" w systemie wykonawczym byłoby praktycznie niewykonalne. Dlatego też w skład pełnego oprogramowania CX-DMOS wchodzi system uruchomieniowy, będący emulacją systemu wykonawczego o następujących cechach:
- jednozadaniowość,
- emulacja funkcji terminala na ekranie,
- systemowa kontrola zasobów bazy danych, realizowana niezależnie od zwykłego dostępu do tych danych,
- praca w środowisku DOS.

Takie podejście umożliwia użycie dodatkowych narzędzi uruchomieniowych, pracujących w środowisku DOS (np. symbolicznych programów uruchomieniowych, właściwych dla środowiska programowania).

System CX-DMOS dysponuje dwoma środowiskami programowania – w języku C oraz w języku CX-BASE. Programy generowania plików definicyjnych są dla obu środowisk wspólne, co wynika z ewolucji, jakim podlegała koncepcja programowania w systemie CX-DMOS.

Pierwotne użycie języka C jako podstawowego było naturalną konsekwencją napisania większości systemu (prócz jądra) właśnie w tym języku. Wybór języka C był podyktowany argumentami merytorycznymi, wynikającymi z faktu, że C w dowolnej implementacji jest zawsze złożeniem struktur języka i bibliotek. Owe argumenty to:
- łatwość implementowania i modyfikowania bibliotek niestandardowych, dostosowanych do wymagań systemu CX-DMOS,
- ułatwione przenoszenie programów do systemów opartych na innym niż DOS jądrze systemowym,
- mały kod wynikowy programu,
- możliwość uruchomienia programów z użyciem programów uruchomieniowych typu CodeView.

Pierwsze doświadczenia w pisaniu programów użytkowych pokazały, że stworzone środowisko jest dość złożone. Co więcej, język C nie dysponuje wszystkimi pożądanymi typami zmiennych (np. brak zmiennych *BCD* i łańcuchów alfanumerycznych wg polskiej sekwencji znakowej), a realizowane funkcje dostępu do plików bazy danych są niskiego poziomu.

W tej sytuacji wprowadzenie wyższego stopnia organizacji przetwarzania danych stało się koniecznością. Zrealizowano to definiując proceduralny język programowania baz danych CX-Base oraz jego pełne środowisko.

JĘZYK CX-Base

W wielu aspektach CX-Base jest całkiem zwykłym językiem programowania, zawierającym wiele instrukcji sterujących oraz rozbudowany aparat wyrażeń z automatyczną (sygnalizowaną!) konwersją typów. Dane, którymi operują programy, są rekordami plików fizycznych. Dostępne typy danych to:
- typy podstawowe (*char, uchar, int, uint, long, ulong*),
- *BCD* (stosowany w obliczeniach finansowych by nie tracić dokładności w operacjach zmiennoprzecinkowych),
- data/czas,
- tablice jednowymiarowe.

Program w języku CX-Base składa się z definicji danych i specyfikacji plików fizycznych oraz z części wykonawczej, będącej zbiorem procedur. Komunikacja między procedurami jest zapewniona zarówno przez zmienne globalne, jak i parametrycznie (przekazywane przez wartość lub – na żądanie – przez odwołanie).

Specjalny rodzaj procedur, zwanych transakcjami, umożliwia wykonanie złożonych operacji na plikach fizycznych w sposób niepodzielny, wykluczając możliwość wystąpienia blokady. W normalnych procedurach operacje dostępu do plików blokują dostęp do pliku tylko na czas trwania jednej operacji (lub ulegają chwilowemu zawieszeniu, gdy plik jest w użyciu). W transakcjach natomiast wyłączność dostępu do plików jest rezerwowana dla programu użytkowego na początku procedury, a zwalniana bezpośrednio po jego zakończeniu. Oznacza to, że podczas trwania transakcji żaden inny program użytkowy nie ma dostępu do zadanych plików, a ściślej, nie może wykonywać określonych operacji na tych plikach.

W realizacji CX-Base szczególnie starannie opracowano obsługę wyjątków (którą niewiele języków programowania w ogóle dostrzega!). Specjalna instrukcja *HANDLER* pozwala zdefiniować akcję dla każdej z poniższych klas błędów osobno:

PROGRAM	– błąd wymuszony programem,
DDS	– błąd dostępu do słownika bazy danych,
TRANS	– błąd obsługi transakcji,
MASK	– błąd przetwarzania formatek,
ISAM	– błąd modułu *ISAM*,
RANDOM	– błąd obsługi pliku typu *RANDON*,
STREAM	– błąd obsługi pliku typu *BYTESTREAM*,
EXPR	– błąd rozwinięcia wyrażenia,
FILE	– wszystkie błędy dostępu do plików,
ALL	– wszystkie rodzaje błędów.

Zakres widzialności instrukcji *HANDLER* jest lokalny dla danej procedury i procedur przez nią wywoływanych – o ile instrukcja ta nie jest w nich przesłonięta analogiczną deklaracją. Jeśli jednocześnie obowiązują deklaracje dla pokrewnych klas wyjątków, np. *FILE* i *ISAM*, w wypadku zaistnienia błędu, skutek odniesie instrukcja bardziej szczegółowa, a więc w tej sytuacji *ISAM*.

Dodatkowym mechanizmem obsługi sytuacji wyjątkowych są klauzule zawarte w niektórych instrukcjach: *ATEND, ONERROR* i *ONBREAK*. Mają one inny zakres widzialności (ograniczony do jednej instrukcji) w porównaniu do instrukcji *HANDLER*. Szczególną rolę odgrywa klauzula *ATEND*, która pozwala w prosty sposób stwierdzić obecność określonego rekordu w bazie danych.

Język CX-Base zawiera również mechanizm współpracy z procedurami napisanymi w C. W ten sposób mogą być realizowane operacje, których zapis w CX-Base jest trudny bądź niemożliwy. Przykładem takich operacji może być przetwarzanie plików zapisanych w innych formatach niż zrozumiałe dla CX-DMOS. Dane i procedury języka CX-Base są dostępne na poziomie procedur C.

Kluczowymi instrukcjami języka są instrukcje służące do odczytu i zapisu plików oraz współpracy z terminalem i drukarką. Oto kilka z nich (dają one pogląd na zakres dostępnych operacji):

```
READ     plik_dowolnego_typu          ; czytaj
READNEXT plik_indeksowo_sekwencyjny   ; czytaj następny rekord
SETKEY   plik_indeksowo_sekwencyjny   ; ustaw nowy klucz
SETREL   plik_indeksowo_sekwencyjny   ; ustaw nową relację
UPDATE   plik_indeksowo_sekwencyjny   ; aktualizuj
REWRITE  plik_indeksowo_sekwencyjny   ; aktualizuj w miejscu
SPLIT    plik_listowy                 ; podziel stronę na dwie
PRINT    zestaw_formatek_drukarkowych ; drukuj
```

```
ACCEPT   zestaw_formatek_ekranowych ; czytaj dane z terminala
MESSAGE  komunikat                  ; pisz komunikat na
                                      terminal
```

Ponadto język CX-Base dysponuje bogatą biblioteką, zawierającą 110 funkcji.

Programy napisane w języku CX-Base podlegają dwuprzebiegowej kompilacji, której wynikiem jest specyficzny kod pośredni. Kompilacja zapewnia weryfikację poprawności semantycznej i syntaktycznej, współpracując z plikami definicyjnymi. Pozwala to uprościć pracę programisty, nie ma bowiem np. potrzeby deklarowania w programie pól plików fizycznych, gdyż nazwy i opisy są pobierane z pliku słownikowego *CONFIG.DDS*. Kod pośredni jest następnie interpretowany przez wydzielony podsystem systemu CX-DMOS.

PRZYGOTOWANIE PROGRAMÓW

Kolejne etapy tworzenia programów użytkowych można zapisać następująco:

1. Przygotowanie struktur danych dla programu użytkowego za pomocą narzędzi opisanych dalej *REGEN, EDEN, EDEP*.
2. Zapis algorytmu przetwarzania.
3. Przygotowanie wynikowych plików systemowych.
4. Przetestowanie programu pod kontrolą programu uruchomieniowego.
5. Uruchomienie programu użytkowego w pełnym systemie.

Schemat ten przedstawiono precyzyjniej na rysunku 1. A oto krótka charakterystyka kilku z wymienionych powyżej programów narzędziowych.

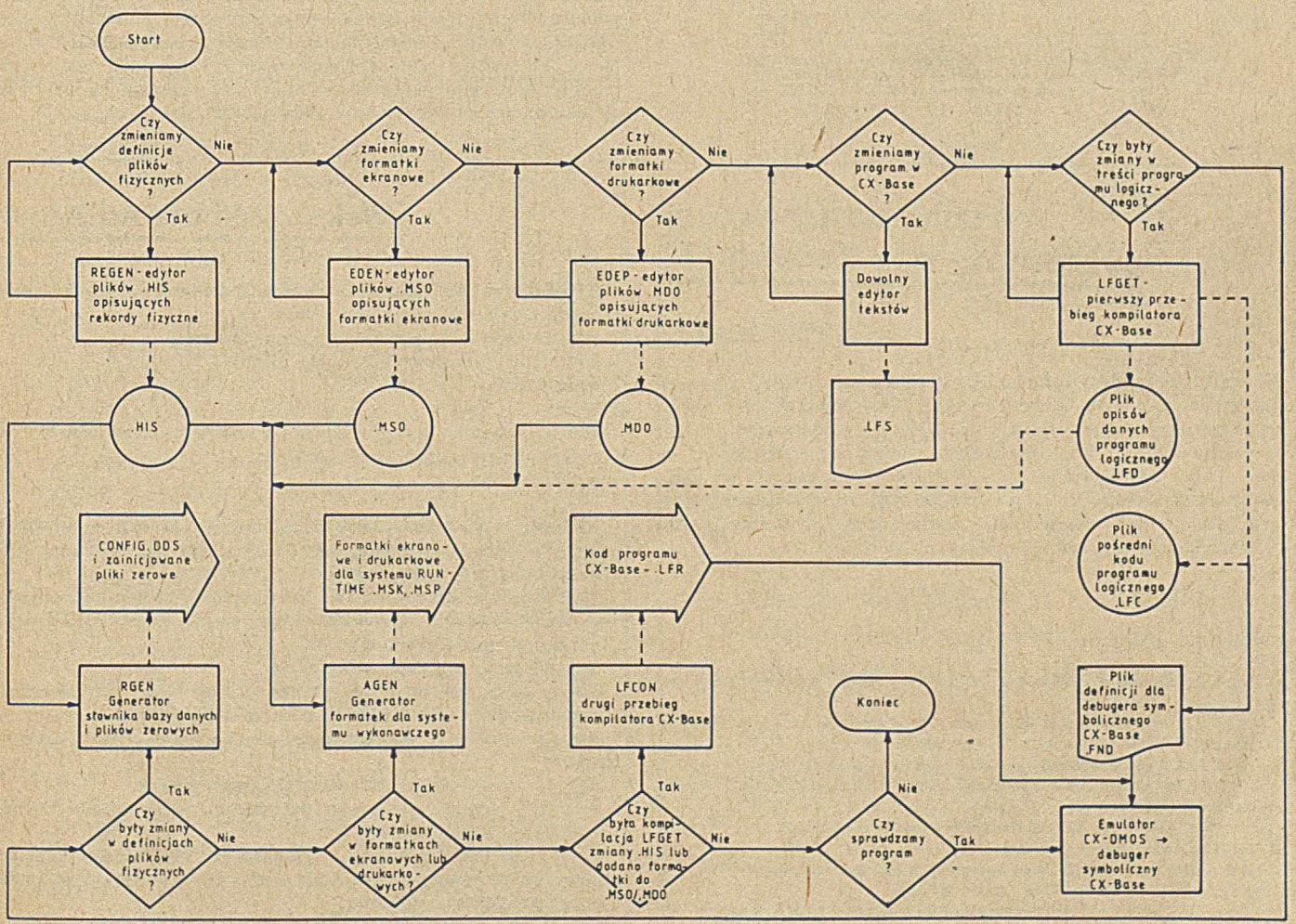
REGEN.EXE umożliwia zdefiniowanie rekordów plików, a więc typów i długości pól oraz kluczy dostępu. Podobnie jak we wszystkich pozostałych programach narzędziowych, definiowanie parametrów polega na wybraniu jednej z wartości podpowiadanych przez program.

EDEN.EXE służy do tworzenia formatów wyświetlania i wprowadzania danych na terminalu. Jest on najbardziej złożonym z tej grupy programów. Na rysunku 2 pokazano obraz przykładowego określania parametrów jednego z pól ekranowych, a na rysunku 3 – obraz określania powiązań logicznych pól ekranu i pól rekordów plików fizycznych. W głównym oknie jest umieszczona lista dostępnych buforów plików bazy danych. W dwóch dolnych oknach znajdują się: spis pól ekranowych i spis pól rekordu wybranego pliku bazy danych.

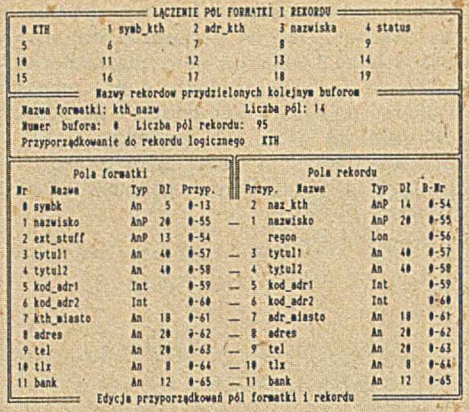
```
PARAMETRY POLA FORMATKI

Pole nr 0   Długosc pola: 5   Pole jest juz ustawione !!!

NAZWA POLA                    symbk
TYP POLA                      alfanumeryczne
KOMUNIKACJA ZE ZMIENNĄ        pole inicjowane spacjami
POLE ZEWNĘTRZNE               —
POLE DO WYSZUKIWANIA          —
DZWONEK PRZY WEJSCIU DO POLA  —
AUTORETURN                    —
POLE TAJNE                    —
WYMAGANE WPROWADZENIE ZNAKU   —
WYMAGANE WYPEŁNIENIE POLA     —
DOSUWANIE W POLU              bez dosuwania
KONWERSJA NA DUZE LITERY      TAK
WYŁĄCZNOŚĆ WPROWADZANIA CYFR  —
POLE KODOWANE W PASCII        —
AUTO-BREAK                    TAK
BREAK BEZWARUNKOWY            —

strzałki Home - zmiana pozycji, « » i SP - zmiana parametru, ESC - wyjście.
F3-akcept., F4=F3+nast., F5=F3+nast.nieust., F9/F10-poprz./nast.pole.
```

Rys. 2

Narzędziem o olbrzymich możliwościach w uruchamianiu programów użytkowych jest symboliczny program uruchomieniowy *LFEXD.EXE*. Podobnie jak w innych programach systemu uruchomieniowego, informacje pomocnicze („ściągawka"), są dostępne po naciśnięciu klawisza *F*1. Efekt wywołania pokazano na rys. 4.

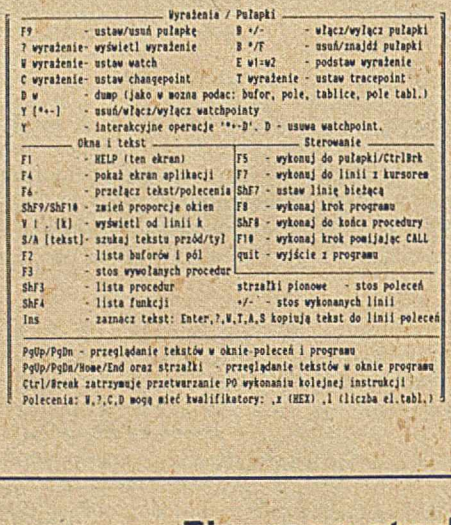


Rys. 1. Kolejność działań w procesie tworzenia aplikacji w języku CX-Base pod kontrolą systemu CX-DMOS



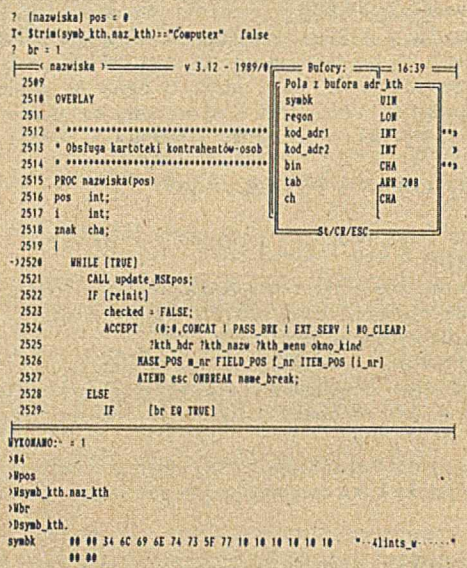
Rys. 3

Pod wieloma względami program ten rozwija filozofię CodeView firmy MicroSoft. Na szczególną uwagę zasługują bardzo pomocne pułapki typu *changepoint* i *tracepoint*, które w połączeniu ze stosem ostatnio wykonanych 100 wierszy programu stanowią bardzo mocne narzędzie uruchomieniowe.



Rys. 4

Na rysunku 5 przedstawiono rzut ekranu sesji programu *LFEXD* w trybie 35-wierszowym. W górnym oknie jest wyświetlana wartość śledzonych zmiennych i punkt ewentualnego przerwania. W oknie środkowym jest umieszczony właściwy tekst uruchomienionego programu z wyświetloną listą plików oraz buforem jednego z nich. Okno dolne jest przeznaczone na konwersację z programistą.



Rys. 5

Autorzy artykułu mają pełną świadomość, że przedstawiony opis środowiska CX-DMOS jest jednocześnie i dość obszerny, i bardzo pobieżny. Narzucone to zostało samym rozmiarem tematu, przy chęci jego całościowego zaprezentowania. Tytułem zakończenia warto wspomnieć, że system i jego aplikacje (kadry, księgowość, rozliczenia z odbiorcami, gospodarka materiałowa, płace) zostały już wdrożone w kilku dużych przedsiębiorstwach krajowych.

Fizyczne struktury systemów baz danych (1)

dokończenie ze s. 5

W tym celu w każdym węźle umieszcza się tablicę o liczbie elementów równej liczbie znaków alfabetu (rys. 15). Porównywany znak klucza będącego argumentem wyszukiwania służy jako indeks w tej tablicy, dzięki czemu nie jest konieczne sekwencyjne jej przeszukiwanie. Duża efektywność wyszukiwania trie drzew jest więc okupiona ich dużą pamięciochłonnością. Ich stosowanie jest najbardziej opłacalne wówczas, gdy liczność zbioru kluczy pamiętanych jest porównywalna z licznością zbioru wszystkich możliwych kluczy.

LITERATURA

[1] Aho A.V., Hopcroft J.E., Ullman J.D.: Projektowanie i analiza algorytmów komputerowych. PWN, Warszawa 1983
[2] Bentley J.L.: Multidimensional Binary Search Trees Used for Associative Searching. Communications of the ACM, Vol. 18, No 9, pp. 509–517, 1975
[3] Date C.J.: A Guide to INGRES. Addison Westley Reading (MA), 1987
[4] Knuth D.: The Art of Computer Programming. Vol. 3: Sorting and Searching. Addison Westley, Reading (MA), 1973
[5] Korth H.F., Silberschatz A.: Database System Concepts. McGrow-Hill
[6] Martin J.: Organizacja baz danych. PWN, Warszawa 1983
[7] Nicklas B.M.: Schlageter G.: Index Structuring in Inverted Data Bases by Tries. The Computer Journal, Vol. 20, No. 4, pp. 321–324, 1977
[8] Reingold E., Nievergelt J., Deo N.: Algorytmy kombinatoryczne. PWN, Warszawa 1985
[9] Sibuja M., Jamamoto T.: Algoritmy obrabotki dannych (tłum. z jęz. japońskiego). MIR, Moskwa 1986
[10] Ullman J.D.: Systemy baz danych. WNT, Warszawa 1988.

Wskazówki dla Autorów

Nadsyłane artykuły nie mogą być publikowane lub przeznaczone do opublikowania w innych czasopismach.

Materiał oprócz tekstu zasadniczego powinien zawierać:

- krótki życiorys zawodowy Autora i jego zdjęcie,
- wykaz literatury,
- tabele,
- materiał ilustracyjny (rysunki, zdjęcia czarno-białe, wydruki) dołączony do artykułu (nie wklejać materiału w tekst),
- podpisy pod ilustracje.

Na osobnej stronie prosimy podać tytuł naukowy, imię i nazwisko, nazwę zakładu pracy, adres domowy (koniecznie!), numer telefonu oraz informację, jaką drogą przekazać honorarium – kasa Wydawnictwa, poczta, bank.

Tekst artykułu prosimy dostarczyć w formie maszynopisu lub wydruku komputerowego, pisany jednostronnie, z podwójnym odstępem – bardzo ważne! – czyli 30 wierszy na stronie i 60 znaków w wierszu.

Wykaz literatury w porządku alfabetycznym.

Tabele – każda na osobnej stronie – powinny być numerowane i opatrzone tytułem oraz ściśle związane z tekstem (zaznaczone miejsce tabeli w tekście).

Rysunki – winny być czytelne (zaznaczyć ich miejsce w tekście), mogą być wykonane ołówkiem.

Zdjęcia – czarno-białe, kontrastowe, na błyszczącym papierze.

Wydruki – czytelne, kontrastowe, wykonane na białym papierze. Format – maksymalnie 18 cm w podstawie.

Podpisy pod ilustracje powinny być napisane na oddzielnej stronie, oprócz kolejnego numeru powinny zawierać tytuł (rysunku, zdjęcia, wydruku) i ew. legendę dotyczącą poszczególnych elementów.

Autor opublikowanego w INFORMATYCE artykułu otrzymuje honorarium i bezpłatny egzemplarz okazowy.

Materiałów nie zamówionych redakcja nie zwraca.

KARTY PRZETWORNIKÓW ANALOGOWO-CYFROWYCH

do komputerów IBM PC/XT/AT technologia „CONVERT"; podzespoły ANALOG DEVICES

- ☐ do 32 wejść analogowych
- ☐ napięcia wejściowe: ±5V, ±10V, 0–10V, 0–20V
- ☐ dynamika 12 bitów
- ☐ przetwornica DC/DC zasilająca część analogową
- ☐ precyzyjny wzmacniacz pamiętająco-próbkujący
- ☐ dwa analogowe kanały wyjściowe
- ☐ napięcia wyjściowe: ±5V, ±10V, 0–10 V, 0–20V
- ☐ trzy 16-bitowe progromowalne liczniki
- ☐ układ generacji przerwań
- ☐ współpraca z DMA (próbkowanie 'w tle')
- ☐ licznik zdarzeń
- ☐ osiem wejściowych
  oraz osiem wyjściowych kanałów cyfrowych

DO TEGO NIEODPŁATNIE:
OPROGRAMOWANIE PODSTAWOWE oraz POMOC W INSTALACJI

INFORMACJE:
'CONVERT T.S.'
53-143 Wrocław, ul. Orla 2A, Tel. 61-92-28 0/4/90

Docierają do redakcji sygnały z całego kraju o trudnościach z nabyciem INFORMATYKI w kioskach Ruchu.

Pragniemy wyjaśnić Czytelnikom przyczynę tych kłopotów. Redakcja podjęła decyzję zmniejszenia liczby egzemplarzy skierowanych do sprzedaży kioskowej, gdyż Ruch pobiera bardzo wysoką prowizję za swoją usługę, nie gwarantując prawidłowego rozdziału egzemplarzy w sieci kioskowej.

W związku z tym prosimy wszystkich zainteresowanych nabyciem INFORMATYKI o kontakt z redakcją (tel. 39-14-34) lub Działem Handlowym SIGMY (tel. 40-30-86) – będziemy Państwu wysyłać egzemplarze za zaliczeniem pocztowym – oczywiście w miarę naszych możliwości dość ograniczonych wysokością nakładu. Czytelnicy z terenu i okolic Warszawy mogą kupić poszczególne numery w Klubie SIGMY przy ul. Mazowieckiej 12, w księgarni „Domu Książki" przy ul. Mokotowskiej 51/53 oraz w lokalu redakcji.

Jednocześnie przypominamy, że najpewniejszą formą otrzymywania INFORMATYKI jest prenumerata, gdyż ona głównie kształtuje wysokość nakładu. Szczegóły jej zamawiania podajemy na stronie II okładki.

Micomp

ZAKŁAD SYSTEMÓW MIKROKOMPUTEROWYCH MICOMP

40–045 Katowice, ul. Astrów 7
telefon i telefaks: 518-628
teleks: 315687

Systemy teletransmisji danych ICL 1900, 2900, ME 29, 39, ODRA 1305, EMCX 1305:

skaner **MPXSCAN-8007,**
procesor sieci WAN
MICOMP-8075
(emulacja ICL 7503),
program teletransmisji danych
MICROS FXBM
(PC, rozproszone bazy danych, wersja sieciowa – NOVELL),
adaptery, modemy, testery,
emulatory terminali

Integracja systemów ICL i ODRA z mikrokomputerami standardu PC i systemami NOVELL, UNIX SCO

Instalacje w największych systemach na terenie całego kraju!

I.O. 1124 89

Komputery do obliczeń naukowych

Komputery te obejmują szeroki zakres systemów o dużej rozpiętości mocy obliczeniowej, dostosowanych do specyfiki różnych obszarów zastosowań. Należą do szczególnie szybko rozwijającej się części rynku informatycznego. Ocenia się, że do roku 1992 roczny przyrost produkcji tego rodzaju komputerów przekraczać będzie 25%, przy czym dla systemów mniejszych, takich jak skomputeryzowane stanowiska pracy stacji roboczych, wskaźnik ten będzie nieco niższy (23%). W tym samym czasie podaż standardowych dużych komputerów będzie rozwijać się w tempie zaledwie 6% rocznie, a średnich – 9%.

Duża różnorodność systemów komputerowych, jakie mogą być wykorzystywane w zastosowaniach naukowych i technicznych, wymaga wprowadzenia odpowiedniego systemu ich klasyfikacji. Jednym z takich systemów jest klasyfikacja przyjęta przez Ministerstwo Energetyki USA. Wyróżnia ona siedem klas w zależności od mocy obliczeniowej, wyrażonej w milionach operacji zmiennoprzecinkowych na sekundę (MFLOPS – *Millions Floating-Point Operation per Second*) lub w milionach rozkazów na sekundę (MIPS – *Millions Instruction per Second*), oraz od ceny. Systemy, o których tu mówimy, obejmują klasy od III do VII. Dwie pierwsze klasy, to mikrokomputery i minikomputery o mocy obliczeniowej mniejszej od 0,6 MFLOPS. Załączona tabela podaje nazwy i parametry tych klas.

Gdy po raz pierwszy opracowywano tę klasyfikację, klasy VII jeszcze nie było, gdyż systemy takie nie były wówczas osiągalne. Ostatnio w klasie tej pojawiły się superkomputery rodziny Y-MP firmy Cray, SX firmy NEC oraz ETA-10 firmy ETA. Jednocześnie klasę VI podzielono na dwie podklasy. Ostatnio mówi się o pojawieniu w klasie VII również systemów nowopowstałych firm, takich jak Supercomputer Systems, Evans & Southerland i BBN Advanced Computers. Te dwie ostatnie podały już parametry swych modeli, które przy maksymalnych konfiguracjach mają przekraczać 10 MFLOPS! Steve Chen, założyciel firmy Supercomputer Systems, brał udział w opracowaniu superkomputerów Cray, a obecnie obiecuje, iż na początku lat dziewięćdziesiątych zaoferuje system o stukrotnie większej mocy obliczeniowej od aktualnych rozwiązań firmy Cray. Współpracę z nim nawiązała firma IBM, aby zapewnić możliwość powiązań między tym systemem a środowiskiem dużych maszyn IBM.

Dla większości współczesnych zastosowań istotne są komputery klas III–VI. Rozwój w tych klasach zmierza do obniżenia jednostkowego kosztu operacji, poprawy ogólnej wydajności konfiguracji, rozszerzenia obszaru zastosowań oraz do wypracowania połączeń z popularnymi systemami komunikacyjnymi.

Do klasy III zalicza się takie systemy, jak bardziej rozbudowane wersje modeli IBM AS/400, Prime Computer 5000, Data General ECLIPSE MV/20000 oraz DEC VAX 6000. Wśród mniejszych systemów tej klasy można wymienić duże stacje robocze firm Sun, Apollo i Silicon Graphic, sprzedawane w cenie 60–100 tysięcy dolarów, a stosowane do realizacji złożonych zadań związanych z przetwarzaniem obrazów. Pół miliona operacji wektorowych na sekundę pozwala uzyskać np. wizualizację przepływu cieczy w czasie rzeczywistym. Kilka lat temu były to zadania osiągalne wyłącznie za pomocą superkomputerów, natomiast dziś wiele już firm może zaoferować stację roboczą o takich parametrach.

Graficzne stacje robocze stanowią dziś szybko rozwijający się rynek w wyniku rozwoju architektur o zredukowanej liście rozkazów (RISC). W 1990 r. przewiduje się sprzedaż 200 tysięcy tykich systemów, a więc czterokrotnie więcej niż w roku 1988. Dotychczas najbardziej rozpowszechnioną taką stacją była Precision Architecture firmy Hewlett-Packard, ale konkurencja w postaci rozwiązań SPARC (*Scalable Processor Architecture* – architektura procesorów skalowanych) firmy Sun, R2000/R3000 firmy MIPS, 88000 firmy Intel oraz Clipper firmy Intergraph rozwija się bardzo dynamicznie.

Następną grupą w tej klasie są supermikrokomputery dla wielu użytkowników, działające zwykle w sieciach lokalnych. Można tu wymienić modele 825 i 350 rodziny TOWER firmy NCR (o mocy obliczeniowej do 15,4 MIPS) i U7000 firmy Unisys, działające pod nadzorem systemu Unix, oraz HP9000 firmy Hewlett-Packard i HCH firmy Harris. Ceny wahają się od 100 do 700 tys. dolarów. Do tej klasy należą też superminikomputery, takie jak VAX 600 i VAX 88xx, których moc obliczeniowa jest co najmniej siedem razy większa od VAX-a 11/780.

Najnowszą grupą w tej klasie są mikrosuperkomputery np. rodziny 5000 i 6000 firmy Concurrent lub Nighthawk firmy Harris. Oferują one moc obliczeniową w granicach 6–30 MIPS i do 14 MFLOPS, a ceny prostszych konfiguracji nie przekraczają 100 tys. dolarów. Z reguły stosowane są tu wysokowydajne mikroprocesory 32-bitowe, takie jak MC68030 lub 80386 i 80486 z układami przyspieszającymi operacje zmiennoprzecinkowe. Ich architektura została zapożyczona z mikrokomputerów (magistrale VMEbus lub Multibus). Pracują one pod nadzorem systemu Unix i nie wymagają specjalnych warunków pracy. Przewiduje się znaczne rozszerzenie podaży tej grupy w latach dziewięćdziesiątych.

Klasa IV (duże komputery) jest pewną anomalią, gdyż za osiągnięcie 2–6 MFLOPS płaci się drożej niż za niektóre systemy grupy V. Niestety dużych komputerów nie zalicza się do klas wyższych, ponieważ ich średnie moce obliczeniowe nie osiągają odpowiedniego poziomu. Można tu wymienić systemy takie, jak IBM ES/3090 VF, Unisys 1190, National Advanced Systems AS/EX, Control Data Cyber 990 i Amdahl 5990.

Komputery klasy V przeżywały szybki wzrost w połowie lat osiemdziesiątych. Pierwszym minisuperkomputerem był C-1 firmy Convex Computer z roku 1985. Potem pojawiły się FX firmy Alliant, Symmetry firmy Sequent Computer Systems, Trace firmy Multiflow i System 6400 firmy ELXSI. Do tej klasy zaliczane są również pierwsze superkomputery, takie jak Cray 1 i Cyber 205.

Do klasy V zaliczane są ponadto różne rozwiązania układów przyspieszających, przeznaczone do dołączania do istniejących systemów komputerowych. Zapewniają one zwiększenie szybkości działań zmiennoprzecinkowych i przetwarzania macierzy w tradycyjnych dużych komputerach i minikomputerach. Na przykład, układ M64 firmy Floating Point Systems może być wykorzystany zarówno do DEC VAX, IBM System/370 jak i Unisys 1100. Wykorzystując złożony sprzęt zawierający specjalne rejestry, układy potokowe (ang. *pipeline*) i skrytkowe (ang. *cache*) oraz optymalizując kompilatory języków, można osiągnąć szybkość 100 MFLOPS, a nawet większą. Wśród firm oferujących takie rozwiązania można jeszcze wymienić Star Technologies, Analogic, Encore i Numerix.

Klasa VI i VII to superkomputery, stanowiące największe systemy obliczeniowe w świecie. Nabywców takich systemów jest na całym świecie zaledwie około 500. Są to głównie duże organizacje badawcze i agencje rządowe, co gwarantuje pokrycie znacznych kosztów związanych z zakupem systemów o bardzo dużej mocy obliczeniowej. Superkomputery pracują najczęściej w trybie wsadowym w sieciach obejmujących znaczne obszary geograficzne. Umożliwia to dogodny dostęp do zasobów informacyjnych i olbrzymiej mocy obliczeniowej pracownikom naukowym tych placówek badawczych, których nie stać na zakup i

Klasyfikacja komputerów wg Ministerstwa Energetyki USA

Klasa	Rodzaj systemu	Szybkość działania		Cena (mln $)
		MFLOPS	MIPS	
VII	Duże superkomputery	60	10–100	20
VI.5	Superkomputery	35–60	10–20	10–20
VI	Superkomputery	20–35	2–10	5–10
V	Minisuperkomputery	6–20	2	0,3–5
IV	Duże komputery	2–6	10	1–13
III	Superminikomputery	0,6–2	1–12	0,01–1

eksploatację własnego superkomputera, natomiast instalacjom superkomputerowym dzięki opłatom użytkowników zapewnia nie tylko zwrot poniesionych nakładów, ale czasem również zysk.

Koszt jest tu bowiem istotnym czynnikiem ze względu na wysoką cenę, zarówno instalacji jak i eksploatacji. Dotyczy to nie tylko wartości bezwzględnej, ale również kosztu za wykonanie miliona operacji zmiennoprzecinkowych, który dla mniejszych systemów wynosi poniżej 10 tys. dolarów, natomiast w klasie VI i VII sięga 20–40 tysięcy. Jednakże superkomputery pozwalają na rozwiązanie problemów, które w inny sposób byłyby nieosiągalne. Komputery takie stały się niezastąpione w takich dziedzinach jak: astronomia, dynamika płynów, fizyka molekularna i badania genetyczne. Ponadto w wielu zastosowaniach użycie superkomputerów okazało się najtańszym rozwiązaniem, np. w przypadku symulacji tuneli aerodynamicznych.

Tłum. i oprac.
J. RYŻKO
Na podstawie Faulkner Technical Reports, styczeń 1990 r.

Technologie układów scalonych w latach dziewięćdziesiątych

Dotychczas w dziedzinie bardzo szybkich układów scalonych konkurowały ze sobą układy ECL (*Emitter Coupled Logic* – układy o sprzężeniu emiterowym), najszybsze dotąd układy logiczne, z układami zbudowanymi z arsenku galu. Te ostatnie są wprawdzie około dwukrotnie szybsze (kilkadziesiąt pikosekund w porównaniu ze stu kilkudziesięcioma dla ECL), lecz realizowane dla mniejszej (też około dwukrotnie) liczby komórek. Szybki rozwój układów BiCMOS, które wykorzystują zalety układów bipolarnych i CMOS (*Complimentary Metal Oxide Semiconductor* – dopełniające układy jednopolarne typu metal – tlenek), występujących w tej samej kostce, może spowodować eliminację układów ECL lub powstanie połączonej technologii ECL/BiCMOS. Postęp jaki w tej dziedzinie osiągnięto w 1989 r. spowodował, że 24 producentów obwodów scalonych ogłosiło parametry swych nowych układów BiCMOS, zarówno standardowych jak i wytwarzanych według częściowych specyfikacji użytkownika. Parametry te są od dwóch do dziesięciu razy lepsze od tych, jakie uzyskuje się w aktualnych układach bipolarnych lub CMOS. Na przykład, firma Vanguard Semiconductor z Miltipas opracowała wzmacniacz utrzymywania ścieżki (ang. *track/hold amplifier*), który przy kilkakrotnie niższej cenie jest, jeśli chodzi o szybkość i moc, o rząd wielkości korzystniejszy od istniejących realizacji bipolarnych oraz o 40% lepszy od realizacji hybrydowej.

Układy BiCMOS pojawiają się w różnych dziedzinach obejmując mikroprocesory, układy wzbudzające magistrale, konwertery analogowo-cyfrowe, wzmacniacze, sterowniki stacji dyskowych, pamięci operacyjne, programowalne układy logiczne, matryce bramkowe i komórki standardowe. Są one na ogół tańsze tak, że w zakresie standardowych układów ASIC spowodują iż albo układy ECL muszą osiągnąć wyższą szybkość i gęstość upakowania, bądź zostaną zastąpione przez układy BiCMOS. Na przykład, układy ECL ASIC (Application Specific Integrated Circuits – układy scalone do specjalnych zastrzeżeń, zob. Informatyka nr 11–12 z 1988 r. s. 40), firmy National Semiconductor zawierają po 50–100 tys. bramek przy opóźnieniach 120 ps, lecz mocy rozpraszanej 30–40 W, co wymaga specjalnego systemu chłodzenia. Natomiast matryce bramkowe BiCMOS złożone z około 50 tys. bramek rozpraszają tylko 10–20 W, chociaż opóźnienia są tu większe i sięgają 500 ps.

Poprawiają się parametry techniczno-ekonomiczne układów z arsenku galu, zwłaszcza w dziedzinie pamięci statycznych i programowalnych układów logicznych. Można tu wymienić układy programowalne firmy Gazelle Microelectronics – o opóźnieniu 5,5 ns i cenie 36 dolarów, pamięci dynamiczne firmy Vitesse Semiconductor i procesory SPARC (*Scalable Processor Architecture* – architektura procesorów skalowanych) firmy System Process and Engineering. Wśród układów o częściowej specyfikacji użytkownika ta sama firma Vitesse oferuje matryce bramkowe GaAs zawierające 30 tys. bramek i układów komórek standardowych liczący 35 tys. bramek, przy mocy rozpraszanej 20 W i szybkości rzędu 300 MHz do 2,5 GHz. Tylko ta firma potrafiła uporać się z defektami w płytkach arsenku galu. Stosowanie tu tranzystorów polowych pozwala na zwiększenie upakowania. Jest to szczególnie widoczne w układach o dużym stopniu scalenia. Z drugiej strony odbywa się ciągły postęp w technologii CMOS. Trzy firmy (National Semiconductor, Plessey Semiconductors i VLSI Technology) oferują matryce złożone z ponad 200 tys. bramek, a przewidywania mówią o 400 tys. w 1991 r., co pozwoli na realizację jednego procesora 68040 i dwu 80386 w pojedynczej matrycy.

Zwiększająca się złożoność projektowania wymaga jednocześnie bardzo drogich narzędzi, co powoduje, że koszt dodatkowych masek staje się pomijalny. Dlatego największe perspektywy mają kombinowane układy ECL i BiCMOS realizowane w urządzeniach łączących własności matryc bramkowych i komórek standardowych. **(JR)**

Macintosh w wersji „laptop"

Firma Wallaby Systems Inc., z Boulder w stanie Colorado, oferuje od lutego 1990 r. system *„nakolanowy"* (ang. *laptop*), kompatybilny z Macintoshem, który będzie można zakupić przez sieć sprzedaży firmy Apple. System ten stanowi rozszerzenie sprzętowe systemów Macintosh Plus lub SE, a jego dołączenie zwiększa ponad dwukrotnie prędkość obliczeniową systemu głównego oraz pojemność pamięci operacyjnej (o 4 MB) i pamięci dyskowej (o 40 MB). Dołączony sprzęt waży około 4,5 kg, a jego cena waha się od 2995 dolarów – ze stacją dysków elastycznych o pojemności 800 KB – do 4295 dolarów z dyskiem 40 MB. Jest to dwukrotnie mniej niż przenośna wersja Macintosha, która waży 8 kg i znajduje się w sprzedaży od jesieni 1989 r.

Rozwój systemów z procesorem 80486

W marcowym numerze czasopisma *Personal Computing* podane są przewidywania (wg opracowania firmy International Data Corp.) rozwoju systemów zbudowanych na mikroprocesorze 80486. W 1989 r. wprowadzono do eksploatacji ok. 2000 takich systemów. W roku 1990 przewiduje się tu kilkudziesięciokrotny wzrost do poziomu ok. 175–200 tys., a w roku 1991 dalszy, ponad dwukrotny wzrost, do około pół miliona systemów. W roku 1992 wzrost ten ma być już tylko rzędu 60–70%, tzn. do ok. 800–850 tysięcy.

Kostki DRAM Simensa

Firma Siemens AG z Monachium rozpoczęła masową produkcję kostek pamięci dynamicznej o pojemności 4 Mbitów, na które istnieje już znaczne zapotrzebowanie. Ich ceny mają być zbliżone do aktualnie obowiązujących na rynku światowym. Układ ten ma symbol HYB514100 i jest wynikiem programu badawczego Mega, który firma podjęła wspólnie z holenderską firmą Philips International NV z Eindhoven. Jest on zbudowany na płytce krzemowej o boku 6 cali i zawiera 8,6 mln elementów na powierzchni 91 mm². Zastosowano technologię CMOS oraz kondensatory rowkowe (podstawowy wymiar 0,8 μm). Cykl pracy wynosi 160 ns, a przy pracy ze stroną szybką – 45 ns. Moc rozpraszana wynosi 440 mW przy przetwarzaniu oraz 5,5 mW przy przechowywaniu informacji. Organizacja może być różna: 4 Mbity × 1 bit lub 1 Mbit × 4 bity.

Jednocześnie kontynuowana jest produkcja kostek o pojemności 256 Kb i 1 Mb. W 1989 r. wyprodukowano ponad 50 mln tych układów.

Pierwsza dziesiątka komputerów osobistych

Czasopismo *Personal Computing* z marca br. podało listę dziesięciu modeli komputerów osobistych kupowanych najczęściej przez firmy amerykańskie zatrudniające ponad 500 pracowników. Zestawienie zostało sporządzone dla danych z okresu od sierpnia 1988 do lipca 1989 r.

Z zestawienia widać, że największą popularnością cieszy się rodzina komputerów IBM PS/2, która występuje na tej liście pięciokrotnie, w tym na dwu pierwszych miejscach. Zdecydowaną przewagę uzyskał model 50 tej rodziny, a więc komputer o średniej mocy obliczeniowej (mikroprocesor 80286 o częstotliwości zegara 20 MHz, dysk twardy o pojemności 20 lub 60 MB) i cenie ponad 2000 dolarów. Model ten stanowił aż 15% wszystkich zakupionych komputerów osobistych, a więc ponad dwukrotnie więcej niż zajmujący drugie miejsce na tej liście jeden z najtańszych modeli tej rodziny – PS/2 Model 30 (procesor 8086 z zegarem 8 MHz, dysk twardy o pojemności 20 MB) w cenie ponad 1000 dolarów. Trzecie i czwarte miejsce zajmują z udziałem po 6% Zenith Z248 oraz PS/2 Model 60, oba z procesorem 80286. Z248 ma częstotliwość zegara 12 MHz, pamięć operacyjną o pojemności 1 MB (rozszerzalną do 6 MB), dysk elastyczny 3,5 cala i dysk twardy o pojemności od 40 do 120 MB. PS/2 Model 60 ma dysk twardy 44 lub 70 MB i kosztuje ok. 3300 dolarów.

Następne trzy miejsca (przy udziale po 4%) zajmują: PS/2 Model 70 oraz dwa modele komputera MacIntosh firmy Apple: SE i II. Model 70 ma już procesor 32-bitowy 80386 i dysk twardy 60 lub 120 MB, a jego cena wynosi ok. 3700 dolarów. Komputery MacIntosh są uznane ze swych dobrych własności graficznych i dużych możliwości rozszerzenia konfiguracji.

Na miejscach 8–9 z udziałem po 3% znalazły się: system Deskpro 286 firmy Compaq oraz PS/2 Model 30-286. Pierwszy z nich o częstotliwości zegara 12 MHz i w cenie zaledwie 1500 dolarów, drugi – to zmodernizowana wersja popularnego modelu PS/2, z procesorem 80286, o częstotliwości zegara 10 MHz.

Ostatni w dziesiątce, to system Compaq 386, stanowiący tylko 2% kupionych komputerów, oczywiście z procesorem 80386 i częstotliwościami zegara od 16 do 25 MHz oraz dyskiem twardym od 20 do 300 MB. W konsekwencji tego zróżnicowania rozpiętość cen jest tu również znaczna i wynosi od 2300 do 8200 dolarów.

SPROSTOWANIE

Bardzo przepraszamy **Pana dr. inż. Jacka Malca**, autora artykułu „Maszyny lispowe" za błędne wydrukowanie imienia na s. 7 w nr. 5, 1990 INFORMATYKI.

Redakcja

Systemy 486 w Europie

Główni wytwórcy komputerów w Europie zaczynają wzorem USA oferować komputery osobiste oparte na nowym mikroprocesorze 80486 firmy Intel. Dla firm europejskich jest to szczególnie istotne, ponieważ od tego zależą ich dalsze losy.

Mimo, że przemysł komputerów osobistych nadal rozwija się bardzo dynamicznie (przewidywane 20% wzrostu w 1991 r.) marża zysku zmniejszyła się w ostatnich latach tak znacznie, że zaledwie kilka firm będzie mogło wytrzymać na tym rynku konkurencję. Np. firma General Electric Corp. podała, że najprawdopodobniej nie będzie już prowadziła sprzedaży systemów klasy 486 w Europie ze względu na zbyt wysokie koszty. Natomiast firmy europejskie, takie jak włoska Olivetti, czy niemiecka Siemens będą starały się utrzymać dotychczasową pozycję rynkową, tym bardziej, że sprzedaż systemów bardziej złożonych daje większy zysk niż systemów prostych. Przedsiębiorstwa europejskie muszą jednak stworzyć sieć dystrybucji na całym kontynencie, a nie tylko na tych regionalnych rynkach, gdzie tradycyjnie dominowały. W ten sposób będą mogły stawić czoła wielkim dostawcom amerykańskim, takim jak Compaq Computer Corp. czy Hewlett-Packard Co.

Reputacja firmy Olivetti została jednak już nieco nadwyrężona półtorarocznym opóźnieniem w dostawie zapowiedzianego na rok 1988 komputera osobistego z procesorem 80386. W rezultacie w 1989 r. udział tej firmy w rynku komputerów osobistych zmalał m.in. na korzyść firmy Compaq, która mając 6,2% tego rynku zajmuje obecnie już czwarte miejsce w Europie – za IBM (19,7%), Olivetti (6,8%) i Aplle (6,5%) – a przed firmami Tandom (4,4%) i Commodore (3,3%) Z tej przyczyny właśnie Olivetti szykuje się, aby jako pierwsza z firm europejskich wystąpić z systemami klasy 486, a dysponując własną siecią sklepów ma większe szanse sukcesu niż Compaq, który przy sprzedaży musi korzystać z pośredników. Oferuje on już dwa nowe modele klasy 486 wyposażone w magistralę EISA (*Extended Industry Standard* – rozszerzona standardowa architektura przemysłowa), ale ponieważ w USA nastąpił spadek sprzedaży tych wyrobów, firma liczy na zbyt w Europie aż 40% swej produkcji. Compaq ma już w Szkocji wytwórnię kart oraz zamiar odkupienia tam montowni od Wang Laboratories Inc. Jeden z oferowanych modeli o nazwie *Systempro* jest wprawdzie systemem wieloprocesorowym i wielozadaniowym, lecz wielu specjalistów ocenia, że użytkownicy będą mieli jeszcze kłopoty z jego oprogramowaniem, nie mówiąc już o tym, że stosunkowo wysoka cena znacznie ograniczy krąg potencjalnych nabywców.

Compaq i Olivetti będą miały wkrótce poważnego konkurenta w niemieckiej firmie Siemens, która kupując we Francji wytwórnię komputerów osobistych IN2 weszła na rynki poza niemieckie. Oczekuje się szybkiego ogłoszenia przez Siemensa parametrów jego systemu 486. Innym konkurentem jest Hewlett-Packard, którego Vectra 486 ma również architekturę EISA. Firma ta ma wprawdzie tylko 2,8% europejskiego rynku komputerów osobistych, ale zdobyła już duży rozgłos jako dostawca skomputeryzowanych stanowisk pracy (ang. *workstations*). Wszyscy oczekują też na ofertę w tej klasie sprzętu ze strony IBM, która już wprowadziła dodatkową kartę o symbolu PS/2 486/25 do swych modeli z mikroprocesorem 80386.

Niezależnie jednak od wkroczenia IBM widać wyraźnie, że walka o udział w europejskim rynku komputerów osobistych dotyczyć będzie w znacznej mierze systemów klasy 486, co nastąpi wtedy, gdy nowe kostki Intela będą produkowane w znacznych ilościach. Przewiduje się, że w wyniku tej walki na rynku europejskim pozostanie najwyżej 3–4 miejscowych producentów.

Węgierski mikroprocesor

Brytyjska firma Densitron Ltd. oferuje mikroprocesor o specjalnej architekturze, optymalizowany pod kątem technologii przetwarzania równoległego. Wymaga on specjalnych technik programowania, które firma zamierza uzyskać wykorzystując węgierskich programistów sprowadzanych do Anglii. Kostka wytwarzana jest w technologii HCMOS przy wymiarze podstawowym 1,5 μm. Jej parametry są równoważne układom logicznym typu ECL firmy Motorola z serii 10K. Mnożenie macierzy binarnych o rozmiarach 1000 × 1000 trwa 1,28 s. Główną część mikroprocesora stanowi matryca 8 × 8 komórek procesorowych, z których każda ma swą pamięć wewnętrzną i układy wejścia-wyjścia. Komórki połączone są magistralą przypominającą sieć i pracują ze wspólnym zegarem. Każda z nich ma własną listę rozkazów mikrokodu, co pozwala tworzyć konfigurację dla określonej funkcji przekształcania.

Mikroprocesor można zastosować do budowy komputerów o równoległym przetwarzaniu lub realizacji zadań wymagających masowej manipulacji na bitach, jakie występują przy przetwarzaniu obrazów lub manipulacji na macierzach wektorowych. Koszt kostki wynosi 30 dolarów, a zestawu wspomagającego, który zawiera dodatkową kartę do IBM PC, oprogramowanie symulatora na poziomie bitowym oraz prosty system pracy – 2700 dolarów.

Nowe książki

WYDAWNICTWA NAUKOWO-TECHNICZNE

Witold Sikorski: Wprowadzenie do użytkowania mikrokomputerów. WNT, Warszawa 1989, wyd. II, nakład 15 tys. egz., s. 174, ISBN 83-204-1173-4

W książce przedstawiono języki programowania Fortran, Pascal i Basic oraz ich najbardziej znane translatory. Ponadto opisano najczęściej stosowane programy usługowe m.in. edytory tekstowe, bazy danych ze wskazaniem ich zalet i możliwości.

Książka jest przeznaczona dla osób rozpoczynających pracę z mikrokomputerami, a także dla tych, którzy wykorzystują mikrokomputery do prac biurowych.

Enn Ch. Tyugu: Programowanie z bazą wiedzy. WNT, Warszawa 1989, wyd. I, nakład 3 tys. egz., s. 238. ISBN 83-204-0965-9

W książce tej Autor przedstawia metodę programowania (zwaną programowaniem z bazą wiedzy), której istota polega na tym, że programista zamiast opisywać procesy obliczeniowe – buduje bazę wiedzy zawierającą dane i warunki pozwalające rozwiązać postawione zadania. Korzystając z tej informacji, system generuje automatycznie program, który produkuje rozwiązanie – gdy ono istnieje. Metoda ta pozwala na znaczne uproszczenie prac programowych związanych z typowymi, powtarzającymi się zadaniami. Czytelnik znajdzie ogólny opis systemu realizującego przedstawioną metodę (w dostosowanym do niej języku Utopist), a także bardzo liczne przykłady zastosowań, m.in. do typowych zadań matematycznych i fizycznych, budowy baz danych, projektowania systemów itp.

Książka jest przeznaczona dla programistów, pracowników naukowych zajmujących się informatyką i studentów kierunków informatycznych.

Witold Lipski: Kombinatoryka dla programistów. WNT, Warszawa 1989, wyd. II, nakład 4 tys. egz., s. 180, ISBN 83-204-1023-1

W książce przedstawiono wybrane zagadnienia kombinatoryki, teorii grafów i algorytmów kombinatorycznych. Szczególny nacisk położono na alegoryczne podejście do problemów kombinatorycznych. Rozważanym problemem towarzyszą zwykle szczegółowe algorytmy ich rozwiązania i analiza złożoności obliczeniowej tych algorytmów. Algorytmy zamieszczone w pracy to nieco skondensowane nieformalne wersje programów napisanych w Pascalu. Każdy z rozdziałów zawiera na końcu zbiór zadań.

Książka jest przeznaczona dla programistów, projektantów systemów przetwarzania informacji, pracowników nauki (zajmujących się informatyką oraz dla studentów kierunków informatycznych.

Niklaus Wirth: Algorytmy + struktury danych = programy. WNT, Warszawa 1989, wyd. II, nakład 20 tys. egz., s. 374, ISBN 83-204-1120-3

W książce przedstawiono systematyczne podejście do rozwiązywania problemów programistycznych o charakterze zarówno naukowym, jak i praktycznym. W prezentowanych modelach szczególną uwagę zwrócono na zasady strukturalizacji danych w procesie tworzenia programu z uwzględnieniem różnych dziedzin zastosowań. Przedstawiono także algorytmy sortowania, algorytmy rekurencyjne, dynamiczne struktury językowe i kompilatory.

Książka jest przeznaczona dla szerokiego kręgu programistów, użytkowników komputerów i studentów kierunków informatycznych.

E.A. Jakubajtis: Lokalne sieci komputerowe. WNT, Warszawa 1989, wyd. I, nakład 6 tys. egz., s. 208, ISBN 83-204-1079-7

Książka jest poświęcona nowemu typowi sieci komputerowych – sieciom lokalnym. Opisano w niej modele lokalnych sieci komputerowych, ich architekturę oraz strukturę logiczną i fizyczną. Przedstawiono kolejne podsieci komunikacyjne i sposoby dostarczania informacji. Podano przykłady wielu typów sieci oraz omówiono różnego rodzaju urządzenia włączane do sieci lokalnych.

Książka jest przeznaczona dla studentów elektroniki i informatyki oraz inżynierów zajmujących się projektowaniem lokalnych sieci komputerowych.

Jan Bielecki: Wprowadzenie do języka C. WNT, Warszawa 1988, wyd. I, nakład 25 tys. egz., s. 236, ISBN 83-204-0947-0

W książce przedstawiono język C – jeden z najważniejszych współczesnych języków programowania mikrokomputerów. Opis języka zilustrowano licznymi i szczegółowo skomentowanymi przykładami programów. Przytoczono opisy implementacji dokonanej przez firmę HiSoft dla mikrokomputerów ZX Spectrum i Amstrad 6128.

Książka jest przeznaczona dla szerokiego grona Czytelników – użytkowników popularnych mikrokomputerów domowych i profesjonalnych – pragnących się nauczyć programowania w jezyku C.

Słownik informatyki polsko-angielsko-rosyjski – praca zbiorowa. WNT, Warszawa 1989, wydanie III rozszerzone i zmienione, nakład 20 tys. egz., ISBN 8-204-1062-2

Słownik jest przeznaczony dla szerokiego kręgu Czytelników zainteresowanych informatyką i mających z nią do czynienia w swojej pracy. W doborze haseł kierowano się głównie powszechnością ich stosowania. W Słowniku uwzględniono terminologię dotyczącą informacji, jej przetwarzania i zasad działania komputerów. Słownik nie zawiera terminów dotyczących zastosowań komputerów. W pracy znajduje się ok. 2 tys. terminów z definicjami, podanych w układzie alfabetycznym, z odpowiednikami angielskimi i rosyjskimi.

On Wirus GOTO Gdziekolwiek

dokończenie z III s. okładki

Dlaczego tak bardzo ważną decyzją jest znalezienie i pozbycie się z naszego grona tych, którzy w gruncie rzeczy podcinają gałąź na której siedzą. Co więcej, napisanie programu wirusa wymaga ogromnego doświadczenia i wiedzy, a więc lista potencjalnych autorów jest niewielka. Smutne jest, że z reguły są to najlepsi programiści.

Oczywiście nie należę do idealistów, sądzących, że same apele coś zmienią. Zbyt wielka jest pokusa i zbyt słabe, jak na razie, są możliwości wykrycia sprawcy, aby nie pojawiali się kolejni inteligentni programiści tworzący nowe wersje wirusa. Jakie są motywy ich działania? Przecież muszą poświęcić sporo czasu na opracowanie programu, za który raczej nie mogą oczekiwać wynagrodzenia (chyba, że równocześnie zaczną sprzedawać program „antybiotyk"). Motorem ich działania może być więc chęć sprawdzenia się, zrobienia psikusa, czy wreszcie chęć odwetu na pracodawcy, który – na przykład – zwolnił z pracy lub zbyt mało płaci. Właściwie tylko te dwa ostatnie przypadki są groźne.

Nowe przepisy prawa powstają wtedy, gdy społeczeństwo stwierdza istnienie zjawisk, z którymi nie chce się zgodzić. Również i u nas powinno dojść do sformułowania lub interpretacji istniejących przepisów określających świadome rozpowszechnianie wirusów za działalność przestępczą. Zanim jednak to nastąpi (nadmiar ograniczeń prawnych nigdy w rezultacie nie wychodzi na zdrowie społeczeństwu) lepiej jest wytworzyć środowiskowy kodeks honorowy traktujący bardzo poważnie tych, którzy niszczą nasz wspólny dorobek. Żadne usprawiedliwienia nie mogą być tutaj zastosowane. Wprowadziłeś świadomie program wirusa do komputera lub napisałeś wspaniały nowy program wirusa – jesteś łajdak!

*Przy tej okazji, może by zorganizować **Fundusz do Walki z Wirusami Komputerowymi**. Pieniądze zgromadzone na tym Funduszu, byłyby przeznaczone dla tych wszystkich, którzy dostarczą istotnych informacji na temat wykrytych przez nich wirusów oraz tych, którzy opracowują programy antywirusowe. A skąd te pieniądze? Tutaj odwołuję się jeszcze raz do tez przedstawionych w tych zBITkach. Oczywiście każdy, kto realnie skorzysta z programów lub informacji powinien wnieść dotację na ten fundusz. Będzie to na pewno tańsze niż reanimowanie jego komputera. W każdym razie pomysł ten proszę traktować jako jeszcze nieuczesany. Wypada tutaj pochwalić PC Kurier, który stał się już trybuną informującą o złapanych wirusach i o istniejących dla nich „antybiotykach".*

WACŁAW ISZKOWSKI

P.S. Zbitki poglądów i spostrzeżeń, tutaj prezentowanych, nie są oficjalnym stanowiskiem Redakcji ani żadnej innej organizacji. Ich prezentacja ma służyć pobudzeniu dyskusji na dany temat, nawet przez wzburzenie moich ewentualnych adwersarzy.

Muszyński M.: Fizyczne struktury systemów baz danych (1)

INFORMATYKA 1990, nr 8, s. 1

Pierwsza część wprowadzenia w dziedzinę fizycznych struktur systemów baz danych, zawierająca omówienie jednowymiarowych drzewiastych struktur danych.

Muszyński M.: Physical structures of database systems (1)

INFORMATYKA 1990, No.8, p. 1

First part of an introduction to the field of database structures, which includes discussion of data one-dimensional tree structures.

Muszyński M.: Die physischen Strukturen von Datenbanksystemen (1)

INFORMATYKA 1990, Nr. 8, S. 1

Erster Teil einer Einführung im Bereich der physischen Strukturen von Datenbanksystemen, der eine Besprechung von eindimensionalen baumartigen Datenstrukturen umfasst.

Postół M.: Przemysłowe systemy czasu rzeczywistego (1). Rola i funkcje oprogramowania systemowego

INFORMATYKA 1990, nr 8, s. 7

Charakterystyka przemysłowych systemów czasu rzeczywistego oraz propozycja nowej koncepcji ich oprogramowania przez zastosowanie autonomicznej biblioteki modułów języka Modula-2.

Postól M.: Industrial real-time systems (1). Part and functions of the system software

INFORMATYKA 1990, No.8, p. 7

Characteristics of industrial real-time systems and presentation of a new idea proposal for building their software based on application of the Modula-2 modules autonomic library.

Postól M.: Industrielle Echtzeitsysteme (1). Die Rolle und Funktionen der Systemsoftware

INFORMATYKA 1990, Nr. 8, S. 7

Eine Charakteristik der industriellen Echtzeitsysteme und ein Vorschlag des neuen Konzeptes ihrer Software mittels Anwendung der autonomischen Modulbibliothek von Modula-2-Sprache.

Ryżko J., Wrzeszcz Z.: Optyczna pamięć dyskowa

INFORMATYKA 1990, nr 8, s. 10

Charakterystyka różnych rozwiązań technicznych optycznej pamięci dyskowej oraz ich zastosowań i produkcji, a także sytuacji krajowej w tej dziedzinie.

Ryżko J., Wrzeszcz Z.: Optical disk storage

INFORMATYKA 1990, No.8, p. 10

Characteristics of different technical solutions of the optical disk storage and its applications and production, as well as of the home situation in this field.

Ryżko J., Wrzeszcz Z.: Optischer Plattenspeicher

INFORMATYKA 1990, Nr. 8, S. 10

Eine Charakteristik von verschiedenen technischen Lösungen des optischen Plattenspeichers, seiner Anwendungen und Produktion, sowie der inländischen Lage in diesem Bereich.

Rekuć W.: Odwzorowania w relacyjnej bazie danych o nieokreślonych a priori typach

INFORMATYKA 1990, nr 8, s. 15

Charakterystyka programowych rozwiązań systemu informatycznego prowadzących do zmniejszenia jego wrażliwości na zmiany w strukturze bazy danych.

Rekuć W.: Images in relational database with in advance undetermined types

INFORMATYKA 1990, No.8, p. 15

Characteristics of software solutions for a data processing system, which lead to reduction of its sensibility against changes in database structure.

Rekuć W.: Abbilden in einer Relationsdatenbank mit im voraus unbestimmten Typen

INFORMATYKA 1990, Nr. 8, S. 15

Eine Charakteristik von Programmlösungen eines EDV-Systems, die zur Verminderung seiner Empfindlichkeit gegen Veränderungen in Datenbankstruktur führen.

Rudolf W. R.: System autorski JU-LEK

INFORMATYKA 1990, nr 8, s. 18

Charakterystyka systemów autorskich jako narzędzi do tworzenia komputerowych programów edukacyjnych oraz omówienie systemu JU-LEK, opracowanego dla mikrokomputerów ELWRO 800 Junior.

Rudolf W.R.: The JU-LEK author system

INFORMATYKA 1990, No.8, p. 18

Characteristics of author systems as a tool for building teaching computer programs and discussion of the JU--LEK-system, which was designed for the ELWRO 800 Junior microcomputer.

Rudolf W. R.: JU-LEK-Autorensystem

INFORMATYKA 1990, Nr. 8, S. 18

Eine Charakteristik von Autorensystemen als Hilfsmittel zur Entwicklung der Computerlehrprogramme und eine Besprechung des für ELWRO 800 Junior-Mikrorechner erarbeiteten JU-LEK-Systems.

Nguyen Kim Sach: Wzmacnianie obrazów telewizyjnych metodami komputerowego przetwarzania obrazów

INFORMATYKA 1990, nr 8, s. 22

Charakterystyka metod poprawy jakości obrazów telewizyjnych oraz omówienie wyników przeprowadzonych przez autora eksperymentów.

Nguyeen Kim Sach: Intensification of television images using methods of computerized image processing

INFORMATYKA 1990, No.8, p. 22

Characteristics of methods for television images quality improving and discussion of experiment results acquired by the author.

Nguyen Kim Sach: Fernsehbilderverstärkung mit Anwendung von Computerbildverarbeitungsmethoden

INFORMATYKA 1990, Nr. 8, S. 22

Eine Charakteristik von Methode der Qualitätsverbesserung von Fernsehbildern und eine Ergebnissebesprechung der vom Autor durchgeführten Experimente.

Jurczak L., Warski W.: System baz danych CX-DMOS (2)

INFORMATYKA 1990, nr 8, s. 24

Druga część charakterystyki wielodostępnego systemu baz danych CX-DMOS, zawierająca omówienie proceduralnego języka programowania baz danych CX-Base oraz sposobu przygotowania programów użytkowych.

Jurczak L., Warski W.: The CX-DMOS database system (2)

INFORMATYKA 1990, No.8, p. 24

Second part of characteristics of the CX-DMOS multiaccess database system, which includes discussion of the CX-Base, a procedural programming language for databases, as well as of the method for preparation of application programs.

Jurczak L., Warski W.: CX-DMOS-Datenbanksystem (2)

INFORMATYKA 1990, Nr. 8, S. 24

Zweiter Teil einer Charakteristik von CX-DMOS, einem Datenbankmehrbenutzersystem, der eine Besprechung der CX-Base, einer prozeduralen Programmiesprache, sowie einer Methode zur Vorbeitung der Anwendungsprogramme, umfasst.

ON Wirus GOTO Gdziekolwiek

Według nawet ostrożnych szacunków, prawie wszystkie komputery są, były lub będą zarażone kilkoma wirusami równocześnie. Oczywiście nie wszystkie z nich muszą od razu przestać działać. Większość tych wirusów jest żartem sfrustrowanego programisty. Niestety w pozostałych przypadkach żarty już się skończyły. Jeżeli powielanie i następnie nieoczekiwane uruchomienie programu wirusa może spowodować ogromne straty finansowe, opóźnienia w wykonaniu prac, wstrzymanie możliwości nadzorowania i sterowania w czasie rzeczywistym – prowadząc do awarii, katastrofy czy wreszcie śmierci – to wtedy trudno jest spokojnie na to patrzeć. Zresztą 8 maja w naszej Telegazecie ujrzałem informację, że Ministerstwo Obrony USA ogłosiło konkurs na zaprojektowanie wirusa, który drogą łączności radiowej mógłby zostać przesłany do komputerów przeciwnika i tam dokonać paraliżujących spustoszeń.

Mam wrażenie, że podobnie jak w innych sprawach, jesteśmy jeszcze kibicami tego problemu zakładając, że nam niewiele grozi z powodu braku sieci oraz tak dużych i odpowiedzialnych systemów. Niech się więc martwią tam na świecie. Jak wiem, doświadczyliśmy już na własnej mikroskórze kilku wirusów. Mamy nawet specjalistów od ich wykrywania i zwalczania. Nie mamy jednak poczucia rzeczywistego zagrożenia, a przede wszystkim poczucia odpowiedzialności. Słuchając niektórych kolegów programistów mam stale wrażenie, że komputer traktują jeszcze jak kolejkę PIKO dla dorosłych – dobrą intelektualną zabawkę.

Na tej stronie zamieszczono rezolucje Zgromadzenia Ogólnego IFIP (San Francisco 1989) w sprawie wirusów komputerowych. Właściwie niewiele można dodać do treści tego apelu. Pozostaje się jedynie do niego zastosować. Tylko jak?

Oczywiście możemy starać się zlikwidować wirusy na własnym komputerze, którego używamy. Ochrona i obrona jednak kosztuje i to dużo. Konieczne jest bowiem opracowanie lub zakupienie programów potwierdzających istnienie wirusów oraz umożliwiających ewentualne ich niszczenie. Konieczne jest też opracowanie zasad korzystania z komputerów przez obcych użytkowników, a szczególnie zasad składowania archiwów i wprowadzania zbiorów i innych instalacji. Koszty te wielokrotnie wzrastają przy włączeniu komputera do sieci komputerowej. Jedną z przyczyn rozpowszechniania się wirusów u nas w kraju jest też powszechne użytkowanie nielicencjonowanego oprogramowania, kopiowanego z innych komputerów, a nie instalowanego z dyskietek dystrybucyjnych. Przejście na tę prawidłową formę instalacji oprogramowania będzie nas również kosztować – i to nie mało.

Na pytanie, komu przede wszystkim powinno zależeć na ukróceniu rozprzestrzeniania się wirusów, odpowiedź jest łatwa: oczywiście nam informatykom. Wiele minęło lat zanim udało się wyprodukować sprzęt oraz oprogramowanie w miarę niezawodne. Jeszcze więcej wysiłku wymagało przekonanie potencjalnych użytkowników do zastosowania komputerów w ich pracy. Wprowadzenie czynnika niepewności działania systemu niszczy dotychczasowe osiągnięcia, a tym samym może znacznie ograniczyć zapotrzebowanie na naszą pracę. W wielu sytuacjach możemy spotkać klienta rozstrzygającego dylemat: – czy zdecydować się na mało pewny system komputerowy, czy też zrezygnować z komputeryzacji ze względu na wielkość kosztów i brak pewności działania?

Czy więc aby na pewno tego chcieliśmy tłumacząc innym, że informatyka jest kluczem do dobrobytu, a przede wszystkim podstawą nowoczesnego społeczeństwa, jakim chcemy być już teraz?

dokończenie na s. 31

Rezolucja IFIP w sprawie wirusów komputerowych

Zważywszy możliwość ujemnych, a nawet katastrofalnych skutków wprowadzenie wirusów programowych do systemów komputerowych, IFIP nalega:

*** aby wszyscy profesjonalni informatycy na świecie uświadomili sobie niszczycielski potencjał wirusów komputerowych,**

*** aby wszyscy wykładowcy informatyki uświadamiali swym słuchaczom niebezpieczeństwa płynące z występowania wirusów programowych,**

*** aby wszyscy wydawcy powstrzymali się od publikowania szczegółów dotyczących konkretnych wirusów programowych,**

*** aby żaden profesjonalny informatyk nie rozpowszechniał świadomie programów z wirusami, wyjąwszy przypadki legalnych badań w ściśle chronionym środowisku, i aby wszyscy twórcy systemów wykrywania wirusów i ochrony przed wirusami zaprzestali rozpowszechniania testowych programów z wirusami,**

*** aby rządy, uczelnie i producenci komputerów zwiększyli nakłady na badania i tworzenie nowych technik ochrony systemów komputerowych oraz**

*** aby rządy podjęły kroki zmierzające do uznania rozpowszechniania wirusów za działalność przestępczą w rozumieniu kodeksów karnych.**

Zgromadzenie Ogólne zwróciło się do wszystkich organizacji członkowskich IFIP z prośbą o rozpowszechnienie tej rezolucji i o zwrócenie na nią uwagi rządów swych krajów.

(wg Biuletynu PTI nr 2, 19/90)

PRZEDSIĘBIORSTWO
ZASTOSOWAŃ INFORMATYKI

meditronik

oferuje:

- systemy mikrokomputerowe
- programy aplikacyjne dla różnych dziedzin gospodarki (na życzenie wysyłamy katalog)
- poszukiwane komponenty elektroniczne
- interfejs do kamery video (opc. CCD) z bogatą biblioteką oprogramowania
- emulator Z80
- tester układów scalonych i pamięci
- programator BPROM
- asynchroniczny procesor komunikacyjny
- konwerter RS-232 – Centronics

instaluje:

- połączenia międzykomputerowe (XT/AT – ODRA/RIAD/IBM)
- systemy sieciowe (NOVELL)
- systemy wielodostępne (SCO Xenix 286, 386, Unix System V)

Jeżeli jesteś autorem oryginalnego programu aplikacyjnego

- skontaktuj się z nami, będziemy pośredniczyć w sprzedaży Twojego programu, dbając o ochronę Twoich praw autorskich!

Nasz adres:

00–194 Warszawa, ul. Dzika 4
tel. (02) 635–22–63, 635–22–64,
fax (02) 635–21–95
tlx 816075 medi pl

EO/605/89

Ogłoszenie w INFORMATYCE bezbłędną szansą trafienia do właściwego odbiorcy!

Miesięcznik INFORMATYKA jest od 25 lat **jedynym** w Polsce czasopismem naukowo-technicznym poświęconym wszystkim aspektom technologii informatycznej (sprzęt komputerowy, oprogramowanie, zastosowania). Taka pozycja czasopisma sprawia, że INFORMATYKA dociera nie tylko do najszerszych kręgów środowiska profesjonalnego, ale również do wszystkich zainteresowanych aktualnymi możliwościami tej technologii.

Ogłoszenie w INFORMATYCE to najskuteczniejszy sposób pozyskania nowych odbiorców i rozszerzenia rynku zbytu. Gwarantujemy Państwu największe efekty przy najniższych nakładach finansowych, ponieważ **ceny ogłoszeń w INFORMATYCE są znacznie niższe** niż w innych czasopismach:

• cała strona	500 000 zł
• 3/4 strony	450 000 zł
• 2/3 strony	400 000 zł
• 1/2 strony	300 000 zł
• 1/3 strony	200 000 zł
• 1/4 strony	150 000 zł
• 1/8 strony	100 000 zł
• poniżej 1/8 strony	2 000 zł/cm^2

Dopłaty:

• za dodatkowy kolor	+ 30%
• za I i IV stronę okładki	+ 100%
• za II i III stronę okładki	+ 50%

Rabaty:

• za powtórzenie 3–6 razy	– 10%
• za powtórzenie ponad 6 razy	– 20%
• za teksty o charakterze informacji techniczno-handlowej (artykuły)	– 50%

Zapewniamy szybką i terminową realizację ogłoszeń!

Nasz adres: INFORMATYKA, 01-552 Warszawa, Pl. Inwalidów 10, p. 128, tel. 39-14-34